NOVEL DOCUMENT REPRESENTATIONS BASED ON LABELS AND SEQUENTIAL INFORMATION

A Dissertation
Presented to
The Academic Faculty

by

Seungyeon Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2015

NOVEL DOCUMENT REPRESENTATIONS BASED ON LABELS AND SEQUENTIAL INFORMATION

Approved by:

Professor Guy Lebanon, Advisor
College of Computing
Georgia Institute of Technology

Professor Haesun Park, Co-advisor
College of Computing
Georgia Institute of Technology

Professor Irfan Essa
College of Computing
Georgia Institute of Technology

Professor Jacob Eisenstein
College of Computing
Georgia Institute of Technology

Dr. Samy Bengio
Machine Learning
Google Research

Date Approved: 14 July 2015

ACKNOWLEDGEMENTS

I would like to begin this acknowledgement section with my deepest gratitude to my advisors: Guy Lebanon and Haesun Park. Without their endless support, I could not have finished my doctoral study. From Guy, I learned to think as a scientist. He always supported my initiatives and encouraged me. Haesun always inspired me from her mathematical brilliance and insights. I extend my thanks to thesis committee members, Irfan Essa, Jacob Eisenstein, and Samy Bengio for their insightful comments. I also would like to thank my internship mentor Kevin Small for his advising and warm friendship.

I also wish to thank my colleagues. I loved interactions with SMLV lab mates, Joonseok Lee, Krishnakumar Balasubramanian, Joshua V. Dillon, Mingxuan Sun, Fuxin Li, and Yi Mao, in particular their friendship and entertaining research discussions. I also thank Dr. Park's research group and FASTLAB members: Jingu Kim, Dongryeol Lee, Jaegul Choo, Hannah Kim, Parikshit Ram, Nishant Mehta, Ravi Sastry Ganti Mahapatruni, Rundong Du, Da Kuang, and Yunlong He. I will remember their friendliness and joyfulness.

I enjoyed a lot of time hanging out with my friends. They made my doctoral study delightful and gave me strength to overcome this long and challenging study. I sincerely thank all of them especially from Georgia Tech, Seoul National University, and Bundang high school.

I thank Kwanjeong Educational Foundation Scholarship for generous financial support and giving me opportunities to meet fellow researchers from various fields.

I would like to express my deepest gratitude to my parents and sister for their love and support: Jindal Kim, Younhee Choi, Hyeyeon Kim. With all my heart, I

thank them.

Last but not the least, I am deeply indebted to my dearest girlfriend Hyewon Chung. She completes me.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF TABLES . . . . . **viii**

LIST OF FIGURES . . . . . **x**

SUMMARY . . . . . **xiii**

I INTRODUCTION **1**

1.1 Motivation 1

1.2 Two Major Challenges: Sparsity and Sequentiality 2

1.2.1 Sparsity 2

1.2.2 Sequentiality 3

1.3 Thesis Statement 4

1.4 Four Aspects of a Good Representation 5

1.4.1 Reconstruction Quality 5

1.4.2 Discriminative Power 5

1.4.3 Interpretability 5

1.4.4 Efficient Computation 6

1.5 Key Contributions 6

1.6 Overview 7

II RELATED WORK **8**

2.1 Dealing with Sparsity in Documents 8

2.1.1 Lexicographic Approaches 8

2.1.2 A Smoothing Term 9

2.1.3 Dimensionality Reduction 9

2.1.4 Topic Modeling 10

2.1.5 Topic Modeling Variants: Supervised and Temporal Topic Modeling 11

2.1.6 Other Supervised Techniques 11

2.2 Modeling Sequential Dependencies of Documents 12
2.2.1 Local Segment Approaches 12
2.2.2 Deep Learning Approaches 13
2.2.3 Kernel Smoothing Approaches 14
2.3 Modeling Human Emotions 14
2.3.1 Sentiment and Mood Analysis 14
2.3.2 Psychology 15
III EXPLOITING LABEL CHARACTERISTICS 16
3.1 Label Characteristics 16
3.2 Structured Labels 17
3.2.1 Manifold of Human Emotions 17
3.2.2 The Statistical Model 18
3.2.3 Experiments 23
3.2.4 Application 34
3.2.5 Summary and Discussion 37
3.3 Structured and Temporally Dependent Labels 38
3.3.1 Temporal Dynamics of Human Emotions 38
3.3.2 Temporal Dynamics of Binary Sentiment 39
3.3.3 Temporal Dynamics of Multivariate Emotions 50
3.3.4 Summary and Discussion 53
3.4 Chapter Discussion 54
IV EMPLOYING SEQUENTIAL INFORMATION 56
4.1 Sequential Information 56
4.2 Spatial and Temporal Sequential Information 57
4.2.1 Modeling Version-controlled Documents 57
4.2.2 Space-Time Smoothing for version-controlled documents . . . 59
4.2.3 Visualizing Change in Space-Time 62
4.2.4 Edge Detection 64

4.2.5 Segmentation . . . 68
4.2.6 Predicting Future Operations . . . 70
4.2.7 Summary and Discussion . . . 71
4.3 Local Sequential Information . . . 72
4.3.1 Locality of Documents . . . 72
4.3.2 Local Context . . . 74
4.3.3 Local Context Sparse Coding (LCSC) . . . 76
4.3.4 Experiments . . . 81
4.3.5 Summary and Discussion . . . 88
4.4 Chapter Discussion . . . 89
V UNIFIED VIEW OF UTILIZING BOTH LABELS AND SEQUENTIAL INFORMATION . . . 91
5.1 Labels and Sequential Information . . . 91
5.2 Supervised Local Topic Modeling . . . 91
5.2.1 Local Context Sparse Coding Model . . . 91
5.2.2 Unified Formulation . . . 92
5.2.3 Estimation . . . 93
5.2.4 Experiment . . . 95
5.3 Chapter Discussion . . . 99
VI CONCLUSION . . . 101
6.1 Thesis Summary . . . 101
6.2 Possible Future Directions . . . 102
6.3 Concluding Remarks . . . 103
REFERENCES . . . 104

LIST OF TABLES

1 Macro F1 score and accuracy over the test set in multiclass emotion classification over top 32 moods (top) and over 7 clusters from Figure 4 (bottom). Bold text represent statistically significant (t-test) improvement by using the mood manifold over the corresponding classification method in the original feature space. 32

2 F1 and accuracy over test-set in sentiment polarity task (top): {cheerful, happy, amused} vs {sad, annoyed, depressed, confused}, and detecting energy level (bottom) {sick, exhausted, tired} vs. {curious, amused}. Bold text represent statistically significant (t-test) improvement by using the mood manifold over the corresponding classification method in the original feature space. 33

3 Polarity assignments of moods 44

4 Test set F1 and accuracy results for predicting sentiment polarity and corresponding training time of each method. Bold face shows statistically significant improvement over other competitors (t-test, 95% confidence). 46

5 Test set F1 and accuracy results for predicting emotion among 64 emotions. Bold face shows statistically significant improvement over other competitors (t-test, 95% confidence). See text for details. 52

6 Test set error rate and F1 measure for edge prediction (section boundaries in Wikipedia articles and author change in Google Wave). The space-time domain Ω was divided to a grid with each cell labeled edge ($y = 1$) or no edge ($y = 0$) depending on whether it contained any edges. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to the TextTiling test segmentation algorithm [21] without paragraph boundaries information. Method c corresponds to a logistic regression classifier whose feature set is composed of statistical summaries (mean, median, max, min) of $\dot{\gamma}_s(s,t)$ within the grid cell in question as well as neighboring cells. 67

7 Error rate and F1 measure over held out test set of predicting future UNDO operation in Wikipedia articles. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to a logistic regression based on the term frequency vector of the current version. Method c corresponds a logistic regression that uses summaries (mean, median, max, min) of $\|\dot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_s(s,t)\|$, $g(t)$, and $h(s)$. 70

8 Top words of selected topics using LCSC on a Wikipedia article "Paris." See text for details. 84

9 Comparision of test set classification accuracy for various methods on 5 classes (comp.*) and full 20 classes (*) of 20 newsgroup dataset . . 87

10 Test set classification accuracy on 5 classes (comp.*) of 20 newsgroup dataset for various dictionary sizes (K) and methods. 96

11 Comparision of test set classification accuracy for various methods on WebKB4, a subset of 20 newsgroup dataset (comp.*), and the full set 97

12 Average training time (sec) of estimating β using Block Principle Pivoting (BPP) algorithm [25] and Greedy Coordinate Descent (GCD) method in Algorithm 2 98

LIST OF FIGURES

1 The two-dimensional structure of emotions from [82]. We can interpret top-left to bottom-right axis as expressing sentiment polarity and the top-right to bottom-left axis as expressing engagement. 25

2 Mood centroids $\mathsf{E}(Z|Y = y)$ on the two most prominent dimensions in emotion space fitted from blog posts. The horizontal dimension corresponds to sentiments polarity and the vertical dimension corresponds to mental engagement level (compare with Figure. 1). 25

3 Tessellation of the space spanned by the first two dimensions of mood manifold with 15 "super-emotion" clusters ($\arg\max_y p(Z|Y = t)$). . . 27

4 Dendrogram of moods using complete linkage function on Bhattacharyya distances between moods. The leaves are cut in 15 clusters to reduce clutters. 28

5 Projected centroids of each review score (higher is better) of movie reviews and restaurant reviews on the mood manifold. Both review start from the left side (negative sentiment in mood manifold) and continues to the right side (positive sentiment) with two different unique patterns. 34

6 Test set mean squared error and its improvements on movie review (two figures on the top) and restaurant review (two figures on the bottom) as a function of the sentiment train set size. Prediction using the combined features outperforms the baseline (regression on document space) and the advantage is larger on smaller training set. 35

7 Graphical model of the temporal sentiment analysis model. X denote sequence of documents, T is the corresponding authoring time, Y is for the sentiment, and Z is the continuous latent variable. See text for details. 40

8 Fourier components of the latent variable in the global model. There are three significant periodic components representing the periods: 8 hours, 12 hours, and 24 hours (circadian rhythm). 48

9 Hourly pattern of global sentiment and selected authors. The y axis correspond to the latent variable of the model (higher values correspond to stronger positive sentiment). See text for details. 49

10 Hourly trends of global model and selected authors on the first two dimensions of the manifold (smoothed). Gray words show $\mathsf{E}[Z|Y = y]$. The arrow shows the start of the day (12am) and direction of the progression of each circle. There is clear separation between day and night time. 52

11 Four space-time representations of a simple synthetic version-controlled document over $V = \{1, 2\}$ (see text for more details). The left panel displays the first component of (20) (non-smoothed array of unit vectors corresponding to words). The second and third panels display $[\gamma(s,t)]_1$ for the non-normalized and normalized representations respectively. The fourth panel displays the gradient vector field $(\dot{\gamma}_s(s,t), \dot{\gamma}_t(s,t))$ (contour levels represent the gradient magnitude). The black portions of the first two panels correspond to zero padding due to unequal lengths of the different versions. 61

12 Gradient and edges for a portion of the version controlled Wikipedia Religion article. The left panel displays $\|\dot{\gamma}_s(s,t)\|^2$ (amount of change across document locations for different versions). The second panel displays $\|\dot{\gamma}_t(s,t)\|^2$ (amount of change across versions for different document positions). The third panel displays the local maxima of $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|^2$ which correspond to potential edges, either vertical lines (section and subsection boundaries) or horizontal lines (between substantial revisions). The fourth panel displays boundaries of sections and subsections as black and gray lines respectively. 64

13 Gradient and edges of a portion of the version controlled Atlanta Wikipedia article (top row) and the Google Wave Amazon Kindle FAQ (bottom row). The left column displays the magnitude of the gradient in both space and time $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|$. The middle column displays the local maxima of the gradient magnitude (left column). The right column displays the actual segment boundaries as vertical lines (section headings for Wikipedia and author change in Google Wave). The gradient maxima corresponding to vertical lines in the middle column matches nicely the Wikipedia section boundaries. The gradient maxima corresponding to horizontal lines in the middle column correspond nicely to major revisions indicated by a discontinuities in the location of the section boundaries. 65

14 Predicted segmentation (top) and ground truth segment boundaries (bottom) of portions of the version controlled Wikipedia articles Religion (left), Atlanta (middle) and the Google Wave Amazon Kindle FAQ(right). The predicted segments match the ground truth segment boundaries. Note that the first 100 revisions are used in Google Wave result. The proportion of the segments that appeared in the beginning is keep decreasing while the revisions increases and new segments appears. 69

15 Graphical model of local context sparse coding. z denotes a document representation, and ϕ denotes a local context in a document of length L. D is a shared dictionary (topics), and β is a latent representation of a corresponding local context using D. See Section 4.3.3.1 for details. 76

16 Result of LCSC on the synthetic example of Section 4.3.4.1 in a simplex, each corner of which represents the probability of one of the corresponding character. Filled shapes (Dz) denote document representations on the simplex; unfilled shapes (ϕ) are for local contexts of each document; filled squares are for two topics D_1, D_2. We see clear separation between $\{Dz_1, Dz_2\}$ vs $\{Dz_3, Dz_4\}$. 82

17 Topic assignments at each position of Wikipedia article "Paris" by LDA (top) and LCSC (bottom). The leftmost edge indicates the beginning of the document and the rightmost edge for the end. Each line type indicates a single topic with its vertical position as a corresponding topic strength. LCSC topics are more locally distributed than LDA. Numbers on the bottom figure indicate topic IDs; Table 8 has the detail of each topic. 83

18 Test set classification accuracies with various dictionary sizes (K) and methods (different line styles) 85

19 Test set classification accuracies of LCSC with various smoothing bandwidths (h) 87

SUMMARY

A wide variety of text analysis applications are based on statistical machine learning techniques. The success of those applications is critically affected by how we represent a document. Learning an efficient document representation has two major challenges: sparsity and sequentiality. The sparsity often causes high estimation error, and text's sequential nature, interdependency between words, causes even more complication.

This thesis presents novel document representations to overcome the two challenges. First, I employ label characteristics to estimate a compact document representation. Because label attributes implicitly describe the geometry of dense subspace that has substantial impact, I can effectively resolve the sparsity issue while only focusing the compact subspace. Second, while modeling a document as a joint or conditional distribution between words and their sequential information, I can efficiently reflect sequential nature of text in my document representations. Lastly, the thesis is concluded with a document representation that employs both labels and sequential information in a unified formulation.

The following four criteria are utilized to evaluate the *goodness* of representations: how close a representation is to its original data, how strongly a representation can be distinguished from each other, how easy to interpret a representation by a human, and how much computational effort is needed for a representation.

While pursuing those *good representation* criteria, I was able to obtain document representations that are closer to the original data, stronger in discrimination, and

easier to be understood than traditional document representations. Efficient computation algorithms make the proposed approaches largely scalable. This thesis examines emotion prediction, temporal emotion analysis, modeling documents with edit histories, locally coherent topic modeling, and text categorization tasks for possible applications.

CHAPTER I

INTRODUCTION

1.1 Motivation

The field of representation learning has attracted much attention in machine learning communities as it plays critical roles in recent large scale data analysis. Representations influence various application performance significantly as well as computation burden. Hence, numerous techniques are being discussed in wide communities from classical statistics to deep learning. For example, topic modeling and clustering can be considered as a type of representation learning as they learn a latent representation from raw inputs. Besides, deep learning communities, which gained enormous attention recently, utilize neural networks in order to learn interactions between hierarchies of representations.

In this dissertation, I will focus on document representations because textual data is the most rapidly growing and widely used form of data. Especially, user-generatd contents from social media are overwhelmingly growing and contain underlying fruitful information. Document understanding is becoming the most significant factor of modeling Internet users, products, and social trends. As a result, this particular task is focused by various research communities such as machine learning, natural language processing, or information retrieval communities.

Unlike other data formats, characteristics of textual data are dominated by two major uniquenesses, which makes representation learning challenging. First, textual data is usually in extremely high dimensions with limited number of observations. It makes our observation very sparse, and often leads to high statistical estimation

error. Second, textual data inherently contains sequential dependencies that substantially alter text's semantics. Although sequential modeling is required for accurate document understanding, it is a significantly harder task. Sequential dependencies differentiate two same words that are in different contexts, which leads to much sparser observation. Handling the two characteristics or challenges is the key in efficient document representation learning.

1.2 Two Major Challenges: Sparsity and Sequentiality

1.2.1 Sparsity

The sparsity of textual data is inherently from a characteristic of human language; that is flourishing vocabulary. Most of the time, we do not repeat the same word over and over in a single document. Hence, we do not have as many observations of a single word as varieties of words. Even more, grammatical conjugations enlarge the diversity of vocabulary further.

Sparsity issues are very frequently observed in texts [47]. For example, a famous dataset RCV1 [43] has 40,000 possible terms with average 200 words per a document. With the traditional term-document matrix representation, 99.95% of the matrix is filled in zero. Moreover, social network data, which is rapidly growing recently, is generally much shorter than most standard text datasets. For example, Twitter has a restriction of 150 characters that generally holds only 20 - 30 words in a document.

This widespread and extreme sparsity often causes high estimation error. We simply do not have enough data to estimate the geometry of the event space. Extremely scarce probabilistic mass prevents discovering useful patterns and often perturbs the patterns severely with noisy observations. For example, a linear regression model that uses a term-document matrix X will result in an ill-posed normal equation because the matrix is mostly filled in zero. The solution will be seriously unstable because the matrix is close to singular. Another example would be clustering documents with

Euclidean distance metric. Since we do not have enough shared words between two documents, most pairwise distances would be very large.

There are numerous attempts to alleviate the sparsity issue. Relevant studies will be covered more in Chapter 2, but hereby we briefly discuss the attempts and compare those with ones in this dissertation. First, there are lexicographic approaches such as stemming or other word-transformation methods, which will reduce the dictionary size effectively for denser space. Second, feature selection techniques remove words that do not affect documents' semantics much (e.g. stop words). Third, topic models and latent space methods actively search for a subspace that we are interested in. While these approaches can be combined seamlessly, this dissertation focuses on the last method.

1.2.2 Sequentiality

A sequential flow of text plays a significant role in the semantics of that text. Each word in a document is loosely or tightly connected to another. A context, an envelop of semantic relationships, manipulates meanings of a word drastically. Because of inter-connectivities independently modeling words in a document does not accurately reflect their semantics.

Sequential dependencies introduce another layer of complexity in representation learning. While extracting hidden relationships, such as coreferences or grammatical dependencies, is already challenging, each interdependency expands the event space further and aggravates the sparsity issue. For example, an n-gram model, a very simple sequential modeling method, exponentially expands its features space by generating permutations of vocabulary. Because of this reason, the n-gram model often fails to perform better than a bag-of-words model that does not include any sequential information.

Many research communities try to solve this challenging problem. A large portion

of natural language processing communities focus on estimating hidden relationships in a document such as grammar parsing, coreference resolution, and dependency parsing. Unlike those exact grammatical resolutions, machine learning, information retrieval, and deep learning communities attempt to model sequential dependencies without estimating exact grammar structures. The famous n-gram model generates local fragments of a document to capture local sequential dependencies. [53] proposed a neural language model based on neighboring words and Long Short Term Memory [22] learns representations directly from sequences of words. Approaches in this dissertation follow along the lines of those ungrammatical approaches and provide more scalable estimation procedures.

1.3 Thesis Statement

This dissertation introduces novel document representations to overcome the two challenges, sparsity and sequentiality. By exploiting label characteristics and employing sequential information, we can efficiently obtain good representations that are close to the original, highly discriminative, easy to interpret, and efficiently computed.

A large number of documents have labels that help us to understand the geometry of our interest. By focusing on the space of interest that is more compact than the original space, we can effectively resolve the sparsity issue. This dissertation applies the technique to an emotion prediction problem, the goal of which is determining an emotion label given a document. The characteristics of human emotions, structured and temporally dependent, are utilized in order to accurately model emotions.

Sequential information providing interdependencies of a document is employed in a unique way in this dissertation. By modeling a joint or conditional distribution of a word and its location, we can learn document representations that are more flexible and compact than traditional approaches. I examine the approach in modeling version-controlled documents and building a locally coherent topic model.

Finally, we discuss a unified representation learning framework based on both labels and sequential information. The new representation attempts to solve the two major challenges in one unified formulation.

1.4 Four Aspects of a Good Representation

[2] is a good survey paper that discusses various aspects of a *good representation*. The aspects widely evaluate a representation from functional properties to interpretability. Similar to their aspects but more concisely, I propose four criteria that are considered *essential* in this dissertation.

1.4.1 Reconstruction Quality

Foremost, a representation should reflect the original data. A document representation needs to reflect the original sequence of words at its best. In this thesis, I focus purely on the expressive power for sequential observations despite the fact there could be undisclosed semantic relationships.

1.4.2 Discriminative Power

Differentiating a document from another or separating a group of documents from another group are crucial in various applications. For example, text categorization, sentiment analysis, information retrieval, and clustering are all based on a distance metric from a document to another. A good representation should promote or demote the distinction between documents based on our need.

1.4.3 Interpretability

A human can read a very limited number of documents. Scaling up human interactions demands a concise and intuitive representation of a document. For example, clustering assignments or topics of a document captures general ideas of the document without reading the document as a whole. For sentiment analysis, we can consider a very compact one-dimensional representation for an instant understanding.

1.4.4 Efficient Computation

Scalability of a representation-learning algorithm cannot be overlooked. Practically, it might be the most critical factor affecting the representation's usage. The scale of texts grows exponentially, and we often discover unforeseen patterns only when we handle the data together in massive scale. From the very beginning of designing a representation, we need to consider a scalable algorithm. Parallel or distributed learning algorithms have become crucial as multicore-distributed systems have been widely deployed.

1.5 Key Contributions

This dissertation makes the following contributions:

1. A new way to learn a document representation that is structured and temporally dependent, which improves sentiment and emotion prediction as well as intuitive understanding of human emotions

2. Efficient learning of the model above using Dirac's delta approximation

3. A new way to model a version-controlled document with a joint probability of a word w, its spatial position s, and temporal position t, which provides a consolidated view of a document development process

4. A new way to capture local features of a document using a conditional distribution of a word w given its position t, which produces locally coherent topics and strong classification performance

5. A new way to learn topics with sparse non-negative matrix factorization (or non-negative sparse coding) by a greedy coordinate descent variant

6. A new way to learn a document representation based on both labels and sequential information, which unifies the evaluation criteria in a single formulation

1.6 Overview

Chapter 2 discusses related studies and compares those with my approaches. In Chapter 3, two approaches exploiting label characteristics are presented in order to resolve the sparsity issue. The following chapter (Chapter 4) examines a novel way to incorporate sequential information in document representations. The two different views are unified in a single formulation in Chapter 5. I conclude the dissertation with the final discussion and remarks in Chapter 6.

CHAPTER II

RELATED WORK

2.1 Dealing with Sparsity in Documents

Since sparsity of textual data has been one of the most serious issues in numerous text applications, a large number of approaches were introduced by wide research communities.

2.1.1 Lexicographic Approaches

While a large number of sparsity-resolving techniques can be augmented, the most popular first-stage methods are lexicographic approaches. The lexicographic methods are often called 'preprocessing' and widely employed because of their effectiveness and light computation loads. The approaches can be regarded as feature engineering techniques that transform or select a subset of features.

Stemming is the most famous approach in this type. It converts grammatical conjugations into its stem word. For example, *is* and *are* are converted to *be*; and *does* and *do* are converted to *do*. Conversions are based on a dictionary or an algorithm. Porter Stemming [64] and Krovetz Stemming [29] are the most popular two stemming algorithms.

Filtering out words that do not affect text's semantics is also very useful in many applications. Especially in information retrieval tasks, removing uninformative common words (called *stop words*) effectively reduces the size of a dictionary. A few examples of the stop words are pronouns, be verbs, and do verbs. SMART [70, 42] is a popular and standard stop word list.

2.1.2 A Smoothing Term

Data mining and information retrieval communities frequently form a term-count or a term-frequency vector for a document representation. In this particular format, the vectors are usually extremely sparse, which causes various numerical issues. Adding a smoothing term that alleviates sparsity is often useful in practice. For example, instead of using a naive term frequency representation, we can use the following:

$$\text{repr}(d) = [\text{tf}(w_1, d, \alpha), \ldots, \text{tf}(w_v, d, \alpha)]^\top \tag{1}$$

$$\text{tf}(w, d, \alpha) = \frac{\text{count}(w, d) + \alpha}{\sum_w [\text{count}(w, d) + \alpha]}, \tag{2}$$

where d is a document ID, w is a word, and $\text{count}(w, d)$ is the word count of w in d. α is a smoothing term to avoid sparse vector representation. Although this approach is useful in some applications, such as computing similarity between two documents, the resulting representation still remains in high dimensional space that makes other approaches, such as document clustering or text categorization, inefficient and not scalable.

2.1.3 Dimensionality Reduction

Unlike feature engineering techniques that focus on characteristics of feature itself, dimensionality reduction techniques actively transform the space while focusing on characteristics of data point distribution. The techniques are widely employed for visualization. [17] is a nice survey that covers most popular dimensionality reduction techniques.

The field of dimensionality reduction has two main approaches: linear and non-linear [17]. Linear methods search for a linear projection that produces a subspace. A few popular linear methods are Principle Component Analysis (PCA), Projection Pursuit, Canonical Correlation Analysis, and Independent Component Analysis. Non-linear methods instead search for non-linear embeddings of data. A few notable methods are Multi-Dimensional Scaling, Self-Organizing Map, Locally Linear

Embedding [67], and t-SNE [76].

Dimensionality reduction techniques, such as PCA, are often employed to generate document representations. However, unlike topic modeling methods, latent dimensions of dimensionality reduction techniques are often obscure. The transformed axes of reduced space usually do not have a clear interpretation while ones of topic modeling usually have a apparent probabilistic meaning. Methods in this dissertation are closer to topic modeling techniques in this point of view.

2.1.4 Topic Modeling

While dimensionality reduction mostly focuses on preserving point-wise relationships, topic modeling concentrates on other interpretable criteria such as probability distributions. [12] is a comprehensive study comparing dimensionality reduction, clustering, and topic modeling.

Topic models assume a hierarchy of latent variables that generates words in documents. For example, Latent Dirichlet Allocation (LDA) [7] is a very popular probabilistic topic model that assumes a Dirichlet distribution of topics on a document. Due to its popularity, several variations have been studied [74, 63, 8]. These probabilistic topic models are usually trained by variational inferences or sampling methods.

Unlike probabilistic models, non-probabilistic topic models are mostly based on matrix factorization methods. For example, Latent Semantic Analysis (LSA) [14] and probabilistic LSA [23] employ Singular Vector Decomposition (SVD). [40] makes use of Non-negative Matrix Factorization (NMF) that has a much better interpretation than SVD. In the study of [88], Sparse Topical Coding introduced a sparsity constraint for sparser topic assignments and obtained good performance on various applications.

Models in this dissertation are closer to the non-probabilistic approaches than probabilistic ones. In Chapter 3, I train discriminant models while most probabilistic topic models are fully generative models. In Chapter 4 and 5, I directly utilize sparse

NMF in order to train compact document representations. Please note proposed models are still based on probabilistic assumptions, but I take a practical approach to reduce overall computation by approximations and relaxations.

2.1.5 Topic Modeling Variants: Supervised and Temporal Topic Modeling

Proposed models in Chapter 3 exploit label characteristics, particularly structural and temporal information, during topic learning processes. The models are similar to supervised or temporal topic models in their goals.

Supervised topic models utilize supervised information (labels) during their topic learning process. For example, [6, 66, 87] are extensions of LDA model that introduce additional latent random variables infusing dependencies between a document and a label.

Temporal topic models extend topic models for time series documents. For example, [5, 80] extended LDA to temporally dependent topics over multiple documents. A few other examples are [83, 24]. The papers [39, 48, 49, 50] explored temporal variations of topics and sentiments within a single document.

Even though models in Chapter 3 have a similar goal of the two variants, they are primarily different in their use of structural information. Additionally, the models introduce a continuous latent space while other models assume a discrete latent space. Lastly, I employ specific time information rather than only ordering between documents.

2.1.6 Other Supervised Techniques

Similar to supervised topic modeling, there are other supervised techniques outside of topic modeling studies. Fisher's Linear Discriminant Analysis is a traditional way to discover a subspace maximizing the separation between two classes. Although the subspace does not have probabilistic interpretations unlike topic models, it is still

useful for generating latent representations that reflect categorical label information. My work deviates from Fisher's LDA because I employ additional label relationships such as structural information and temporal dynamics.

WSABIE [84] is another notable study that jointly learns latent representations from images and their annotations (labels). Although the problem domain and formulations are different from my approaches in Chapter 3, the goal, learning representations based on labels, is similar. The main difference is my explicit use of label attributes such as preserving pairwise centroid distances and temporal dynamics.

In the field of multi-task learning, researchers utilize similarities between multiple prediction tasks for better use of data. Particularly, [9] employs Gaussian Process formulation with a similarity covariance between labels. These approaches are similar to the ones in Chapter 3 for their goal although multi-task learning studies do not explicitly generate latent representations.

2.2 Modeling Sequential Dependencies of Documents

It is very natural to consider the sequential flow of a document during its representation learning because the flow drastically changes the document's semantics. Computational linguistic communities have spent enormous effort analyzing grammatical structures of a document in order to model the document accurately. Machine learning, data mining, and information retrieval communities rather focus on extracting rich sequential features from a document instead of the grammar parsing.

2.2.1 Local Segment Approaches

The n-gram model and its variations are widely employed to capture local segments of documents. The original n-gram model generates new features combining n consecutive words. Since the number of new features is overwhelming especially when n is large, back-up rules or hashing techniques are utilized to prevent issues of sparse high-dimensional space. However, improvements from the techniques are limited.

Word2Vec [53, 52] models, Continuous Bag-of-Words and Skip-Gram, take a different approach that learns a parametric model similar to neural language models [4]. Unlike n-gram models, the models do not generate a enormous feature space. They instead learn vector representations of words and a neural language model jointly. An extension of the approach [34] additionally learns paragraphic representations similar to word representations. Even though these models are still based on n-word segments, they perform much better in capturing meaningful representations and in various application performances. While these models require heavy computation when n is large (e.g. $n > 20$), approaches in Chapters 4 and 5 provide simpler and efficient computation with a kernel smoothing technique.

The models above have a predetermined segment size. On the other hand, some studies automatically determine sizes of segments. For example, [11] introduced hierarchical random variables that jointly solve text segmentation and topic modeling.

2.2.2 Deep Learning Approaches

A Recurrent Neural Network (RNN) is able to handle sequential observations using its recurrent hidden layers. The hidden layers hold previous states of latent representations preserving sequential nature of original input. Particularly, Long Short Term Memory (LSTM) [22] directly achieves the sequential modeling by memorizing an arbitrary length of sequential dependencies although it suffers from heavier computation than other RNN types. There is a recent RNN study [33] that is based on rectified linear units and achieved comparable performance along with simpler computation than LSTM.

Compared to deep learning approaches, this dissertation takes a simpler problem formulation. Similar to internal memory structures of RNN or LSTM, a kernel

smoothing technique estimating $p(w,t)$ or $p(w|t)$ preserves local sequential information. With a wide smoothing kernel, a representation will retain a long-term dependency and a narrow kernel for a short-term dependency. The difference comes from a trade-off between memory flexibility and simplified computation. LSTM is on one extreme because it maximizes the flexibility and my approach is on the other extreme since it weights more on computational efficiency while restricting the memorization at a constant length.

2.2.3 Kernel Smoothing Approaches

[39] proposed another approach to capture the sequential flow of a document. They employed a kernel smoothing technique to model sequential word histograms. The resulting smoothed curve captures a document's sequential flow as a whole. Although they do not mention the following explicitly in the paper, their technique estimates a joint probability of a word and its position, $p(w,t)$, similar to a kernel density estimation. Their idea was also explored in [48, 38].

This dissertation extends this direction in Chapters 4 and 5 that covers various models based on a kernel smoothing. First, I examine multi-level sequentiality modeling for a version-controlled document. Second, I explore a conditional probability, $p(w|t)$, to capture local word proximity, which conveys a new concept combining both local segments and a kernel density estimation.

2.3 Modeling Human Emotions

In Chapter 3, I present efficient document representations for emotion detection problems. Hereby, we discuss related studies in an effort to model human emotions.

2.3.1 Sentiment and Mood Analysis

Sentiment or mood analysis has been a significant research direction in natural language processing community. [61, 46] are good surveys on this field. [54, 19, 27] are

notable since they introduced richer emotions compared to previous binary polarity sentiments. The work described in [73, 65, 56] addressed the task of constructing a useful corpus for emotion analysis.

Temporal emotion variations are also widely studied in this field. [55] investigated emotion trends in general, and [59] compared Twitter sentimental trends with Gallop polls. These models are limited in the sense that they do not include temporal dependencies between documents in their model. [49, 50] are notable as they did consider temporal dependencies in their model, but their dependencies were limited within a single document.

2.3.2 Psychology

Psychological communities have made a major effort to discover relationships between human emotions [68, 69, 71, 82, 81, 75, 32]. Note that *emotion*, *affect*, and *mood* have different meanings in psychology, but I use them here interchangeably. The paper [82] is particularly notable as it introduced the dimensional structures of emotions similar to my approach in Chapter 3. My work deviates from research in psychology in that I construct my model based on a large collection of annotated documents instead of a limited number of human surveys. In addition, my model has much higher dimensionality compared to traditional 2-3 dimensions used in psychology.

Temporal variations of emotions are also widely researched in psychological communities. [44] and [77] described the temporal dynamics of emotions. [57] and [20] explored periodic behavior of emotions, called circadian rhythm.

CHAPTER III

EXPLOITING LABEL CHARACTERISTICS

3.1 Label Characteristics

Previously in Chapter 1, we examined problems of sparsity, which are mainly caused by documents' extreme dimensionality. In order to resolve sparsity problems, we need to reduce dimensionality (see Chapter 2 for related studies). In this chapter, we discuss techniques for addressing sparsity by exploiting label information.

Labels are useful for understanding what we are looking for in a feature space, the space of interest, because a large portion of text applications are based on a prediction task for labels. Irrelevant features for the prediction task are less important. Hence, we can use the label information to reduced dimensionality.

Finding compressed space based on labels is a widely studied technique as described in Chapter 2. For example, Linear Discriminant Analysis seeks a linear projection that is helpful in the prediction. Supervised topic models also search for latent random variables that are useful for generating words and labels. These techniques employ label information as an indicator of a class, but we often can extract more information such as structures or dependencies of labels.

Labels are frequently not just simple categorical values. For example in document categories, we often overlook hierarchical or structural relationships of labels. In a sentiment prediction, we often fail to notice that the labels, human emotions, are temporally correlated. Rich characteristics of labels could be very helpful for an efficient document representation learning.

In this chapter, I will examine efficient document representations for emotion prediction tasks. I will exploit characteristics of the labels, which are attributes of

human emotions, in order to obtain effective representations. In Section 3.2, I will focus on one attribute: emotion structures. In the subsequent section (Section 3.3), I will introduce another attribute, temporal dynamics, in the representation learning.

3.2 Structured Labels

3.2.1 Manifold of Human Emotions

Sentiment analysis predicts the presence of a positive or negative emotion y in a text document x. Despite its successes in industry, sentiment analysis is limited as it flattens the structure of human emotions into a single dimension. "Negative" emotions such as `depressed`, `sad`, and `worried` are mapped to the negative part of the real line. "Positive" emotions such as `happy`, `excited`, and `hopeful` are mapped to the positive part of the real line. Other emotions like `curious`, `thoughtful`, and `tired` are mapped to scalars near 0 or are otherwise ignored. The resulting one dimensional line loses much of the complex structure of human emotions. Note that *emotion*, *affect*, and *mood* have distinguishable meanings in psychology, but we use them here interchangeably.

An alternative that has attracted a few researchers in recent years is to construct a finite collection of emotions and fit a predictive model for each emotion $\{p(y_i|x), i = 1, \ldots, C\}$. A multi-label variation that allows a document to reflect more than a single emotion uses a single model $p(y|x)$ where $y \in \{0, 1\}^C$ is a binary vector corresponding to the presence or absence of emotions. In contrast to sentiment analysis, this approach models the higher order structure of human emotions.

There are several significant difficulties with the above approach. First, it is hard to capture a complex statistical relationship between a large number of binary variables (representing emotions) and a high dimensional vector (representing the document). It is also hard to imagine a reliable procedure for compiling a finite list of all possible human emotions. Finally, it is not clear how to use documents expressing

a certain emotion, for example `tired`, in fitting a model for predicting a similar one, for example `sleepy`. Using labeled documents only in fitting models predicting their denoted labels ignores the relationship among emotions, and is problematic for emotions with only a few annotated.

I propose an alternative approach that models a stochastic relationship between the document X, an emotion label Y (such as `sleepy` or `happy`), and a position on the mood manifold Z. I assume that all the emotional aspects in the documents are captured by the manifold, implying that the emotion label Y can be inferred directly from the projection Z of the document on the manifold, without needing to consult the document again.

The key assumption in constructing the manifold Z is that the spatial relationship between $X|Y = j, j = 1, \ldots, C$ is similar to the spatial relationship between $Z|Y = j, j = 1, \ldots, C$ (see assumption 4 in the next section).

Previous work handles the mood prediction problem as multiclass classification with discrete labels. My work stands out in that it assumes a continuous mood manifold and thus develops an inherently different learning paradigm. My logistic regression baseline is generally considered equivalent or better than the ones in related work using SVM [54, 19], Naive Bayes [72]. [27] exploited a user-supplied emotional hierarchy which is an additional assumption that I do not have.

3.2.2 The Statistical Model

I make the following four modeling assumptions concerning the document X, the discrete emotion label $Y \in \{1, 2, \ldots, C\}$, and the position on the continuous mood manifold $Z \in \mathbb{R}^l$.

1. We have the graphical structure: $X \to Z \to Y$, implying that the emotion label $Y \in \{1, \ldots, C\}$ is independent of the document X given Z.

2. The distribution of $Z \in \mathbb{R}^l$ given a specific emotion label $Y = y$ is Gaussian

$$\{Z|Y = y\} \sim \mathcal{N}(\mu_y, \Sigma_y). \tag{3}$$

3. The distribution of Z given the document X (typically in a bag of words or n-gram representation) is a linear regression model

$$\{Z|X = x\} \sim \mathcal{N}(\theta^\top x, \Sigma_x).$$

4. The distances between the vectors in

$$\{\mathsf{E}(Z|Y = y) : y \in C\}$$

are similar to the corresponding distances in

$$\{\mathsf{E}(X|Y = y) : y \in C\}$$

We make the following observations.

- The first assumption implies that the emotion label Y is simply a discretization of the continuous Z. It is consistent with well known research in psychology (see Section 2) and with random projection theory, which state that it is often possible to approximate high dimensional data by projecting it on a low dimensional continuous space.

- While X, Y are high dimensional and discrete, Z is low dimensional and continuous. This, together with the conditional independence in assumption (1) above, implies a higher degree of accuracy than modeling directly $X \to Y$. Intuitively, the number of parameters is on the order of $\dim(X) + \dim(Y)$ as opposed to $\dim(X) \times \dim(Y)$.

- The Gaussian models in assumptions 2 and 3 are simple, and lead to efficient computational procedures. I also found them to work well in the experiments.

The model may be easily adapted, however, to more complex models such as mixture of Gaussians or non-linear regression models (for example, we experimented with quadratic regression models).

- Assumption 4 suggests that we can estimate $\mathsf{E}(Z|Y = y)$ for all $y \in C$ via multidimensional scaling. MDS finds low dimensional coordinates for a set of points that approximates the spatial relationship between the points in the original high dimensional space.

- The models in assumptions 2 and 3 are statistical and can be estimated from data using maximum likelihood.

- The four assumptions above are essential in the sense that if any one of them is removed, we will not be able to consistently estimate the true model.

3.2.2.1 Fitting Parameters and Using the Model

Motivated by the fourth modeling assumption, we determine the parameters $\mu_y = \mathsf{E}(Z|Y = y), y \in C$ by running multidimensional scaling (MDS) or Kernel PCA on the empirical versions of $\{\mathsf{E}(X|Y = y) : y \in C\}$, which are the class averages $\frac{1}{n_k}\sum_{y^{(i)}=k} x^{(i)}$ (n_k is the number of documents belonging to category k).

We estimate the parameter θ, defining the regression $X \to Z$, by maximizing the likelihood

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \sum_i \log p(y^{(i)}|x^{(i)}) \\ &= \arg\max_{\theta} \sum_i \log \int_Z p(y^{(i)}|z) p_\theta(z|x^{(i)}) dz \\ &= \arg\max_{\theta} \sum_i \log \int_Z p(z|y^{(i)}) \frac{p(y^{(i)}) p_\theta(z|x^{(i)})}{\sum_y p(z|y)p(y)} dz.\end{aligned} \tag{4}$$

The covariance matrices Σ_y of the Gaussians $Z|Y = y$, $y = 1, \ldots, C$ may be estimated by computing the empirical variance of Z values simulated from $p_{\hat{\theta}}(Z|X^{(i)})$

for all documents $X^{(i)}$ possessing the right labels $Y^{(i)} = y$. A more computationally efficient alternative is computing the empirical variance of the most likely $\hat{Z}^{(i)}$ values corresponding to documents possessing the appropriate label $Y^{(i)} = y$:

$$\hat{Z}^{(i)} = \arg\max_z p_{\hat{\theta}}(Z = z|X^{(i)}) = \hat{\theta}^\top X^{(i)}. \tag{5}$$

Given a new test document x, we can predict the most likely emotion with

$$\begin{aligned} \hat{y} &= \arg\max_y \int p(y, z|x)dz \\ &= \arg\max_y \int p(y|z)p_{\hat{\theta}}(z|x)dz. \end{aligned} \tag{6}$$

But in many cases, the distribution $p(Z|X)$ provides more insightful information than the single most likely emotion Y.

3.2.2.2 Approximating High Dimensional Integrals

Some of the equations in the previous section require integrating over $Z \in \mathbb{R}^l$, a computationally difficult task when l is not very low. There are, however, several ways to approximate these integrals in a computationally efficient way.

The most well-known approximation is probably Markov chain Monte Carlo (MCMC). Another alternative is the Laplace approximation. A third alternative is based on approximating the Gaussian pdf with Dirac's delta function, also known as an impulse function, resulting in the approximation

$$\begin{aligned} \int N(z\,;\mu,\Sigma)g(z)\,dz &\approx c(\Sigma)\int \delta(z-\mu)g(z)\,dz \\ &= c(\Sigma)g(\mu). \end{aligned} \tag{7}$$

A similar approximation can also be derived using Laplace's method. Obviously, the approximation quality increases as the variance decreases.

Applying (7) to (4) we get

$$\begin{aligned}\hat{\theta} &\approx \arg\max_{\theta} \sum_i \log \frac{p(y^{(i)}) p_\theta(z^{(i)^*}|x^{(i)})}{\sum_y p(z^{(i)^*}|y)p(y)} \\ &= \arg\max_{\theta} \sum_i \log p_\theta(z^{(i)^*}|x^{(i)}) \end{aligned} \tag{8}$$

where $z^{(i)^*} = \arg\max_z p(z|y^{(i)}) = E(Z|y^{(i)})$, which is equivalent to a least squares regression.

Applying (7) to (6) yields a classification rule

$$\hat{y} \approx \arg\max_y p\left(y \middle| Z = \arg\max_z p_{\hat{\theta}}(z|x)\right). \tag{9}$$

3.2.2.3 Implementation

Since $P(Z|Y = y)$ are Gaussian, the resulting Bayes classifier, which minimizes the classification risk, is the well known quadratic discriminant analysis (assuming $\mathsf{Var}(Z|Y = y)$ depends on y), or the well-known linear discriminant analysis (assuming that $\mathsf{Var}(Z|Y = y)$ does not depend on y).

In estimating the covariance matrices of a Gaussian $P(Z|Y = y)$, it is sometimes assumed that each class has the same covariance matrix, leading to linear discriminant analysis (LDA) as the optimal Bayes classifier. The alternative assumption that the covariance matrices for each class is different leads to quadratic discriminant analysis (QDA) as the optimal Bayes classifier.

I consider both assumptions and three different models for the covariance matrices: full covariance, diagonal covariance, and linear combination of full covariance and spherical covariance (standard regularization technique):

$$\hat{\Sigma}' = (1-\lambda)\hat{\Sigma} + \lambda \left(\sum_{i=1}^{C} \hat{\Sigma}_{ii}\right) I \qquad \text{(LDA)}$$

$$\hat{\Sigma}'_y = (1-\lambda)\hat{\Sigma}_y + \lambda \left(\sum_{i=1}^{C} [\hat{\Sigma}_y]_{ii}\right) I \qquad \text{(QDA)}$$

In either case, I used a C dimensional ambient space (C equals the number of emotions) and the approximation (9).

3.2.2.4 Summary of the Model

A summary of the estimation procedure follows:

1. Estimate $E(Z|Y = y)$ using MDS/KPCA
2. Estimate $\hat{\theta}$ using (4) or (8)
3. Estimate Σ_y for $p(Z|Y = y)$ using empirical averages of simulated $Z^{(i)}$ values, or most likely $\hat{Z}^{(i)}$ value as described in Section 3.2.2.1.

Due to the high dimensionality of X, it may be useful to estimate $\hat{\theta}$ using ridge regression, rather than least squares regression. In this case, we update the estimate $E(Z|Y = y)$ in third stage, based on the ridge estimate $\hat{\theta}$.

One interpretation of the model $X \to Z \to Y$ is that Z forms a sufficient statistic of X for Y. We can thus consider adapting a wide variety of predictive models (for example, logistic regression or SVM) on $Z \mapsto Y$. These discriminative classifiers are trained on $\{(\hat{Z}^{(i)}, Y^{(i)}), i = 1, \ldots, n\}$.

3.2.3 Experiments

3.2.3.1 Datasets

I used crawled Livejournal[1] data as the main dataset. Livejournal is a popular blog service that offers emotion annotation capabilities to the authors. About 20% of the blog posts feature these optional annotations in the form of emoticons. The annotations may be chosen from a pre-defined list of possible emotions, or a novel emotion specified by the author. I crawled 15,910,060 documents and selected 1,346,937 documents featuring the most popular 32 emotion labels (in respect to the number of documents annotated in). It is a significantly larger dataset compare to similar works: 1,000 [72], 346,723 [19] and 345,014 [54] documents.

[1] `http://www.livejournal.com`

I used Indri from the Lemur project[2] to extract term frequency features while tokenizing and stemming (using the Krovetz stemmer) words. As is common in sentiment studies [13, 58, 26], I added new features representing negated words. For example, the phrase "not good" is represented as a token "not-good" rather than as two separate words. This resulted in 43,910 features.

I used L_1-normalization, dividing term frequency matrix by the number of total word appearances in each document, and followed with a square root transformation, turning the Euclidean distance to the Hellinger distance. This multinomial geometry outperforms the Euclidean geometry in a variety of text processing tasks, as described in [30, 37].

Building a model solely based on the engineered term frequency features ignores the structure of a sentence or paragraphs. Using richer sets of feature may improve the model further; however, My contribution is presenting the manifold of emotions.

The document length histogram is close to an exponential distribution, with mean 113.51 words and standard deviation 146.65 words. There are plenty of short documents (520,436) having less than 50 words, but there are also some long documents (39,570) having more than 500 words. The average word length is 8.33 characters.

Two other datasets that I use in experiments are the movie review data [60] and the restaurant review data[3] [18] (using the same preprocessing described above).

3.2.3.2 Comparison with Psychological Models

In this section, I compare the model to Watson and Tellegen's well known psychological model (Figure 1). Figure 2 shows the locations of mood centroids $\mathsf{E}(Z|Y = y)$ on the first two dimensions of the mood manifold. We make the following observations.

1. The horizontal axis expresses a sentiment polarity-like emotion. The left part features emotions such as `sad` and `depressed`, while the right part features

[2] `http://www.lemurproject.org/`
[3] `http://www.cs.cmu.edu/~mehrbod/RR/`

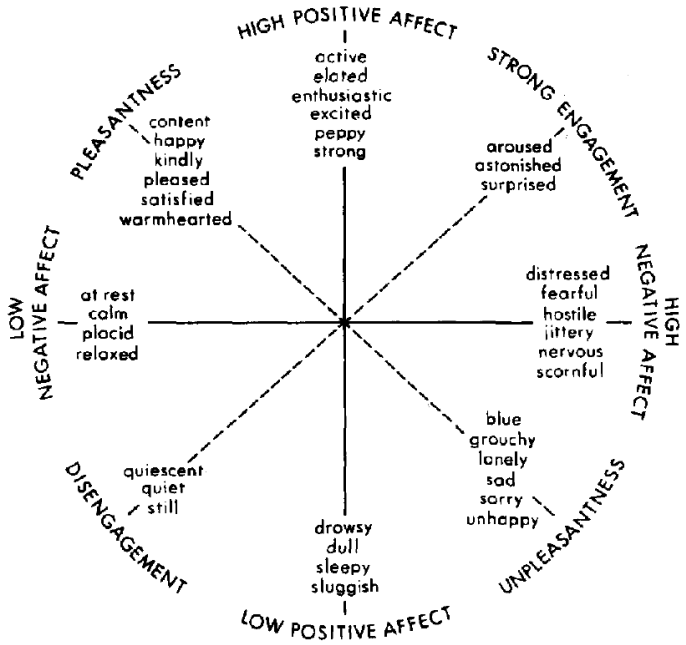

Figure 1: The two-dimensional structure of emotions from [82]. We can interpret top-left to bottom-right axis as expressing sentiment polarity and the top-right to bottom-left axis as expressing engagement.

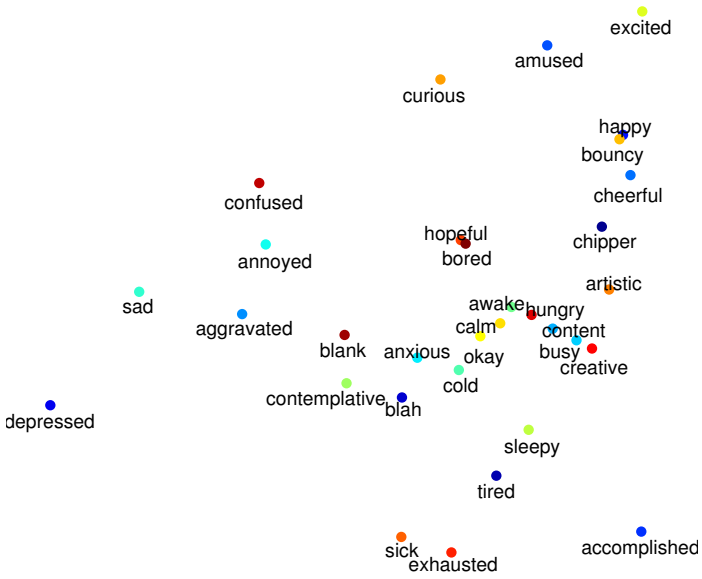

Figure 2: Mood centroids $\mathsf{E}(Z|Y = y)$ on the two most prominent dimensions in emotion space fitted from blog posts. The horizontal dimension corresponds to sentiments polarity and the vertical dimension corresponds to mental engagement level (compare with Figure. 1).

emotions such as `accomplished`, `happy` and `excited`. This is in agreement with Watson and Tellegen's observations (see Figure 1) that identify sentiment polarity as the most prominent factor among human emotions.

2. The vertical axis expresses the level of mental engagement or energy level. The top part features emotions such as `curious` or `excited`, while the bottom part features emotions such as `exhausted` or `tired`. This agrees partially with the engagement dimension in the psychological model. However, the precise definition of engagement seems to be different. For example, in my model (Figure 2), high engagement imply active conscious mental states, such as `curious`, rather than passive emotions such as `astonished` and `surprised` (Figure 1).

3. The neutral moods `blank`, stay in the middle of the picture.

The mood centroid figure is largely intuitive, but the positions of a few centroids is somewhat unintuitive; for example `annoyed` has similar vertical location (energy level) as `bored`. Please note, however, the manifold is higher dimensional and the dimensions beyond the first two provide additional positioning information.

It is interesting to consider the list of words that are most highly scored for each axis in the mood manifold. The words with highest weight associated with the horizontal axis (sentiment polarity) are: `depress, sad, hate, cry, fuck, sigh, died` on the left (negative) side and `excite, awesome, yay, happy, lol, xd, fun` on the right (positive) side. On the vertical axis (energy): `tire, download, exhauste, sleep, sick, finishe, bed` on the bottom side (low energy) and `excite, amuse, laugh, not-wait, hilarious, curious, funny` on the top side (high energy).

I conclude that there is in large part an agreement between the first two dimensions in the model and the standard psychological model. This agreement between the mood manifold and the psychological findings is remarkable in light of the fact that the two models used completely different experimental methodology (blog data vs.

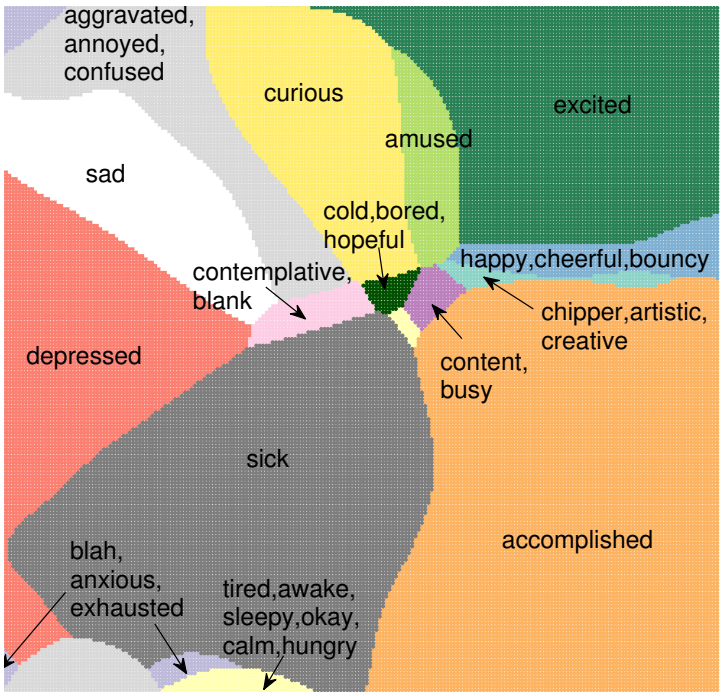

Figure 3: Tessellation of the space spanned by the first two dimensions of mood manifold with 15 "super-emotion" clusters ($\arg\max_y p(Z|Y=t)$).

surveys).

3.2.3.3 Exploring the Emotion Space

Since emotion labels correspond to distributions $P(Z|Y)$, we can cluster these distribution in order to analyze the relationship between the different emotion labels. In the first analysis, we perform hierarchical clustering on the emotions in order to create emotional concepts of varying granularity. This is especially helpful when the original emotions are too fine, (consider for example the two distinct but very similar emotions `annoyed` and `aggravated`). In the second analysis, we visualize the 2D tessellation corresponding to most likely emotions in mood space. This reveals additional information, beyond the centroid locations in Figure 2.

I use the Bhattacharyya dissimilarity,

$$D_B(f,g) = -\log \int \sqrt{f(z)g(z)}dz.$$

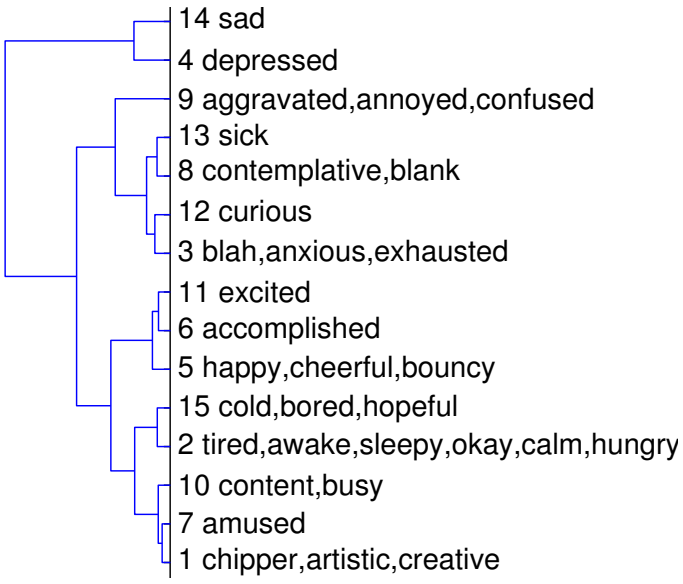

Figure 4: Dendrogram of moods using complete linkage function on Bhattacharyya distances between moods. The leaves are cut in 15 clusters to reduce clutters.

to measure dissimilarity between emotions, which corresponds to the log Hellinger distance between the underlying distributions. In the case of two multivariate Gaussians, it has the following closed form:

$$D_B(N(\mu_1, \Sigma_1), N(\mu_2, \Sigma_2)) = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \log \left(\frac{\det((\Sigma_1 + \Sigma_2)/2)}{\sqrt{\det \Sigma_1 \det \Sigma_2}} \right).$$

Following common practice, I add a small value to the diagonal of the covariance matrices to ensure invertibility.

Figure 4 shows the mood dendrogram obtained by hierarchical clustering of the top 32 emotions using the Bhattacharyya dissimilarity (complete linkage clustering). The bottom part of dendrogram was omitted due to lack of space. The clustering agrees with our intuition in many cases. For example,

1. `aggravated,annoyed` and `confused` are in the same tight cluster.

2. `sad` and `depressed` are very close cluster.

3. `happy`, `cheerful`, and `bouncy` are in the same tight cluster, which is close to `accomplished` and `excited`.

4. `tired, awake, sleepy, okay, calm` and `hungry` are in the same tight cluster.

The hierarchical clustering is useful in aggregating similar emotions. If the situation requires paying attention to one or two "types" of emotions, we can use a particular mood cluster to reflect the desired feature. For example, when analyzing product reviews, we may want to partition the emotions into two clusters: positive and negative. When analyzing the effect of a new advertisement campaign, we may be interested in a clustering based on positive engagement: excited / energetic vs. bored. Other situations may call for other clusters of emotions.

Figure 3 shows the tessellation corresponding to

$$f(z) = \arg\max_{y=1,\ldots,C} p(Z|Y = y).$$

For space and clarity purposes, we use 15 emotion clusters instead of the entire set of 32 emotions. The tessellation shows the regions being classified to each emotion cluster based only on the 2D space. We observe that:

1. As in Figure 2 the horizontal axis corresponds to negative(left) - positive(right) emotion and the vertical axis corresponds to energy level(or engagement): (top) `excited` and `curious` vs. (bottom) `tired` and `exhausted`.

2. The `depressed` region is spread significantly on the left-bottom side, and is neighboring the `sick` region and the `sad` region.

3. The region corresponding to the `happy, cheerful, bouncy` emotions neighbors the `accomplished` region and the `excited` region.

A similar tessellation of a higher dimensional Z space provides additional information. However, visualizing such higher dimensional spaces is substantially harder in paper format.

3.2.3.4 Classifying Emotions

My framework accounts for the relationship between similar and contradictory emotions automatically as it assumes a hidden continuous representation, where $P(Z|Y = y)$ reflects a non-trivial relationship between the emotions. Earlier attempts to construct a manual relationship between emotions based on domain knowledge did not perform well. The current approach is data driven and indeed it outperforms the one vs. all approach, as we show below. My one vs. all baseline is a regularized logistic regression, operating in the original bag of words feature space — one of the strongest text classification baselines.

One of the primary experiment in this work is emotion classification. In other words, given a document x predict the emotion that is expressed in the text. As mentioned in the introduction, this classification can be done by constructing separate $p(y_i|x)$ models for every emotion (one-vs-all approach). However, the one vs. all approach is not entirely satisfactory as it ignores the relationships between similar and contradictory moods. For example, documents labeled as `sleepy` can be helpful when we fit a model for predicting `tired`. The mood manifold provides a natural way to incorporate this information, as documents from similar moods will be mapped to similar points on the manifold.

I performed emotion classification experiment (Table 1, left) on the Livejournal data. I considered the goal of predicting the most popular 32 moods. The class proportion varies in the range 1.72% to 6.52%.

Since 32 moods are too finer in practical usage, I designed coarser classification experiment (Table 1, right) using 7 clusters obtained by hierarchical clustering as in Figure 4. The task is to predict the 7 clusters and cluster proportion varies in the range 4.02% to 28.63%.

I also considered two binary classification tasks (Table 2) obtained by partitioning the set of moods into two clusters (positive vs. negative clusters and high vs. low

energy clusters). The class distributions of these binary tasks are 65.03% vs. 34.97% (sentiment polarity), and 52.17% vs. 47.83% (energy level)

I used half of the data for training and half for testing. To determine statistical significance, I performed t-tests on several random trials. Note that emotion prediction is a hard task, as similar emotions are hard to discriminate (consider for example discriminating between `aggravated` and `annoyed`). It is thus not surprising that prediction performances are relatively low, especially when discriminating between a large number of moods or clusters.

The LDA, QDA and L_2-regularized logistic regression models are implemented in MATLAB (the latter with LBFGS solver). I also regularized the LDA and QDA models by considering multiple models for the covariance matrices. I determined the regularization parameters by examining the performance of the model (on a validation set) on a grid of possible parameter values. I used the same parameters in all experiments.

Table 1 and 2 compare classification results using the original bag of words feature space and the manifold model, using different types of classification methods: LDA, QDA with different covariance matrix models, and logistic regression. Bold faces are improvements over the baseline with statistical significance of t-test of random trials.

Most of experimental results show that the mood manifold model results in statistically significant improvements than using original bag of words feature. Improvements are consistent with various choices of classification methods: LDA, QDA, or logistic regression. The phenomenon is also persistent in variety of tasks: 32 mood classification, more practical 7 cluster classification, or binary tasks. Thus, introducing the mood manifold is indeed made the difference.

I conclude that the manifold contains emotional related information well enough to predict accurate emotional labels. Moreover, improvements over both F1 measure and accuracy indicates low-dimensional manifold indeed help us to train more accurate

Table 1: Macro F1 score and accuracy over the test set in multiclass emotion classification over top 32 moods (top) and over 7 clusters from Figure 4 (bottom). Bold text represent statistically significant (t-test) improvement by using the mood manifold over the corresponding classification method in the original feature space.

		Original Space		Mood Manifold	
		F1	Acc.	F1	Acc.
LDA	full	n/a	n/a	**0.1247**	**0.1635**
	diag.	**0.1229**	0.1441	0.1160	**0.1600**
	spher.	0.0838	0.1075	**0.0896**	**0.1303**
QDA	full	n/a	n/a	**0.1206**	**0.1478**
	diag.	0.0878	0.0931	**0.1118**	**0.1463**
	spher.	0.0777	0.0989	**0.0873**	**0.1253**
Log.Reg.		0.1231	0.1360	**0.1477**	**0.1667**

		Original Space		Mood Manifold	
		F1	Acc.	F1	Acc.
LDA	full	n/a	n/a	**0.2800**	**0.3591**
	diag.	0.2661	0.2890	**0.2806**	**0.3504**
	spher.	0.2056	0.2252	**0.2344**	**0.2876**
QDA	full	n/a	n/a	**0.2506**	**0.3025**
	diag.	0.1869	0.1918	**0.2496**	**0.3088**
	spher.	0.1892	0.2009	**0.2332**	**0.2870**
Log.Reg.		**0.2835**	0.3459	0.2806	**0.3620**

Table 2: F1 and accuracy over test-set in sentiment polarity task (top): {cheerful, happy, amused} vs {sad, annoyed, depressed, confused}, and detecting energy level (bottom) {sick, exhausted, tired} vs. {curious, amused}. Bold text represent statistically significant (t-test) improvement by using the mood manifold over the corresponding classification method in the original feature space.

		Original Space		Mood Manifold	
		F1	Acc.	F1	Acc.
LDA	full	n/a	n/a	**0.7340**	**0.7812**
	diag.	0.7183	0.7436	**0.7365**	**0.7663**
	spher.	0.6358	0.6553	**0.7482**	**0.7699**
QDA	full	n/a	n/a	**0.6500**	**0.7446**
	diag.	0.6390	0.6398	**0.6704**	**0.7510**
	spher.	0.6091	0.6143	**0.7472**	**0.7734**
Log.Reg.		0.7350	0.7624	**0.7509**	**0.7857**

		Original Space		Mood Manifold	
		F1	Acc.	F1	Acc.
LDA	full	n/a	n/a	**0.7084**	**0.7086**
	diag.	0.6441	0.6449	**0.6987**	**0.6989**
	spher.	0.6343	0.6343	**0.6913**	**0.6913**
QDA	full	n/a	n/a	**0.5706**	**0.6100**
	diag.	0.6124	0.6413	**0.6268**	**0.6446**
	spher.	0.6239	0.6294	**0.6754**	**0.6767**
Log.Reg.		0.6694	0.6699	**0.7087**	**0.7089**

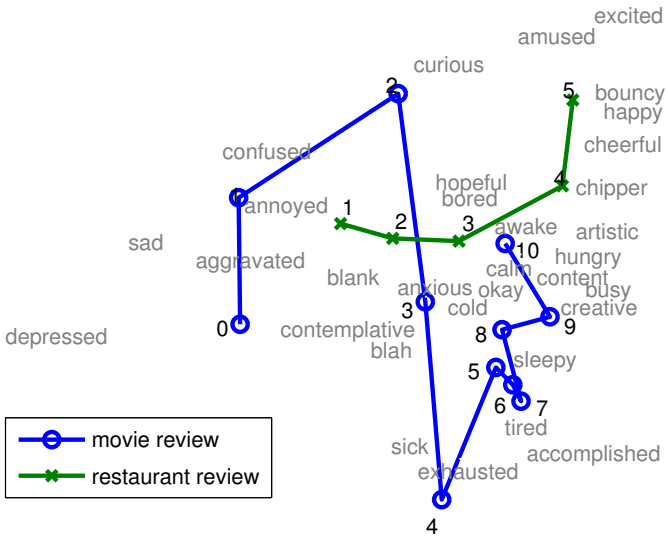

Figure 5: Projected centroids of each review score (higher is better) of movie reviews and restaurant reviews on the mood manifold. Both review start from the left side (negative sentiment in mood manifold) and continues to the right side (positive sentiment) with two different unique patterns.

classifiers.

3.2.4 Application

3.2.4.1 Improving Sentiment Prediction using Mood Manifold

The concept of positive-negative sentiment fits naturally within this framework as it is the first factor in the continuous Z space. Nevertheless, it is unlikely that all sentiment analysis concepts will align perfectly with this dimension. For example, movie reviews and restaurant reviews do not represent identical concepts. In this subsection we visually explore these concepts on the manifold and show that the mood manifold leads to improved sentiment polarity prediction on these domains.

3.2.4.2 Sentiment Notion on the Manifold

I model a sentiment polarity concept as a smooth one dimensional curve within the continuous Z space. As we traverse the curve, we encounter documents corresponding to negative sentiments, changing smoothly into emotions corresponding to positive

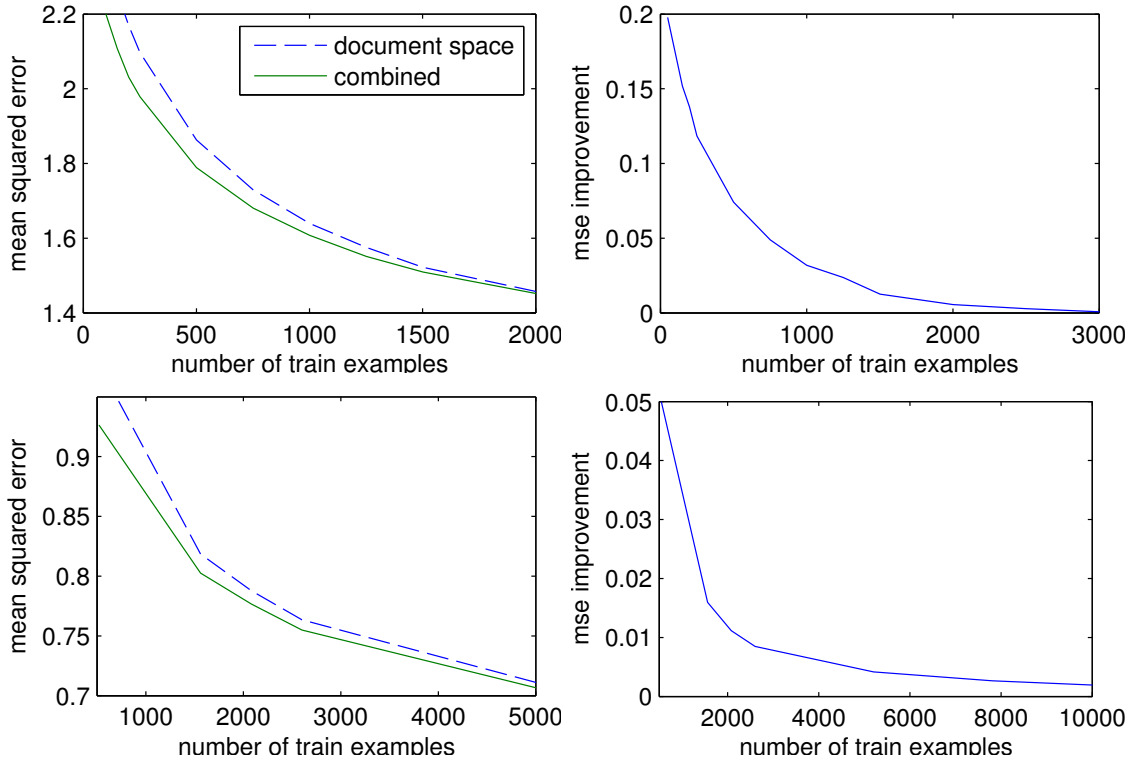

Figure 6: Test set mean squared error and its improvements on movie review (two figures on the top) and restaurant review (two figures on the bottom) as a function of the sentiment train set size. Prediction using the combined features outperforms the baseline (regression on document space) and the advantage is larger on smaller training set.

sentiments. We complement the stochastic embedding $p(Z|X)$ with a smooth probabilistic mapping $\pi(R|Z)$ into the sentiment scale. The prediction rule becomes

$$\hat{r} = \arg\max_r \int p(Z = z|X)\pi(R = r|Z = z)\, dz$$

and its approximated version is

$$\hat{r} = \arg\max_r \pi \left(R = r \Big| Z = \arg\max_z P(Z = z|X) \right)$$

Figure 5 shows the smooth curves corresponding to $\mathsf{E}[\pi(R = r|Z)]$ for movie reviews and restaurant reviews. Both curves progress from the left (negative sentiment) to the right (positive sentiment). But the two curves show a clear distinction: the movie review sentiment concept is in the bottom part of the figure, while the restaurant review sentiment concept is in the top part of the figure. I conclude that positive restaurant reviews exhibit a higher degree of excitement and happiness than positive movie reviews.

3.2.4.3 Improving Sentiment Prediction

The mood manifold captures most of the information for predicting movie review scores or restaurant review scores. Some useful information for review prediction, however, is not captured within the mood manifold. This applies in particular to phrases that are relevant to the review scores, and yet convey no emotional contents. Examples include (in the case of movie reviews) `Oscar`, `Shakespearean`, and `$300M`.

I thus propose to combine the bag of words TF representation with the mood manifold within a linear regression setting. We regularize the model using a group lasso regularization [86], which performs implicit parameter selection by encouraging sparsity

$$\arg\min_w \frac{1}{n} \sum_{i=1}^{n} (w_1^T x^{(i)} + w_2^T z^{(i)} - y^{(i)})^2 + \lambda(||w_1||_2 + \lambda_2||w_2||_2).$$

Above, $z^{(i)}$ is the projection of $x^{(i)}$ on the mood manifold, and λ and λ_2 are regularization parameters. The regularization parameters was determined on performance on validation set and fixed throughout all experiments.

Figure 6 shows the test L_2 prediction error of the method and baseline (ridge regression trained on the original TF features) as a function of the train set size. The group lasso regression performs consistently better than regression on the original features. The advantage obtained from the mood manifold representation decays with the train set size, which is consistent with statistical theory. In other words, when the train set is relatively small, the mood manifold improves sentiment prediction substantially.

I also compared sentiment prediction using the bag of words features and sentiment prediction using the mood manifold exclusively. The mood manifold regression performs better than bag of words regression for smaller train set sizes but worse for larger train set sizes.

3.2.5 Summary and Discussion

I introduced a continuous representation for human emotions and constructed a statistical model connecting it to documents and to a discrete set of emotions. The fitted model bears close similarities to models developed in the psychological literature based on human survey data.

The new document representation in this section improved several aspects of the *good representation* (Section 1.4) while minimizing drawbacks of other aspects.

1. Reconstruction Quality: The representation is a linear subspace from the original document space. Although the manifold loses some information because of the linear projection, it contains all necessary emotion-related information

for emotion predictions. I have proven this by classification experiments comparing the performance of the original space and the one of the manifold (Section 3.2.3.4).

2. Discriminative Power: The mood manifold has stronger discriminative power that showed significantly improved F1 and accuracy (Section 3.2.3.4).

3. Interpretability: In Section 3.2.3.2 and 3.2.3.3, we found the representation is readily understood.

4. Computation: The mood manifold can be learned very efficiently by Dirac's delta approximation (Section 3.2.2.2). The learning process only contains standard linear regressions and empirical Gaussian fittings.

3.3 Structured and Temporally Dependent Labels

3.3.1 Temporal Dynamics of Human Emotions

In the previous section, we discussed sentiment analysis by introducing a multivariate response variable $y \in \mathbb{R}^d$, which corresponds to a more complex emotional state. For example, a classical model from psychology examines a two-dimensional emotional quantity in which the first dimension corresponds to sentiment and the second corresponds to the level of engagement (`aroused` vs. `calm`). Sentiment analysis and its generalizations are important tools in industry, attracting a considerable amount of attention from the research community.

The analysis of human emotions based on text has focused mostly on static analysis, that is the analysis of documents solely based on its own content ignoring temporal dependences. This section explores a temporal model for human emotions that applies to a sequential stream of text documents written by the same author across different time points. My model is somewhat similar to Brownian motion and the Kalman filter and generalizes the latent space emotion model in Section 3.2.

Most papers on sentiment analysis use movie or item reviews, which are poorly suited to temporal modeling. Reviews relate to specific stationary truth and are unlikely to significantly change based on the time of authoring. Blog posts, however, are free expressions of the author's emotions and thus depend more on the time of authoring. Therefore, I demonstrate a temporal model on data consisting of time-stamped blog posts. I construct the sentiment or emotion ground truth from an emotion label for the text.

The temporal model is useful in two ways. First, it leads to a predictive model that estimates the current emotional state of the author within a specific time context. This predictive model is more accurate than static analysis, which ignores time information. Second, the model is useful in confirming or refuting psychological models concerning human emotions and their dependency on time. Specifically, I re-examine the circadian rhythm model from psychology and investigate its higher order generalizations and its variance across multiple individuals.

The work differs from previous studies in primarily three ways. First, unlike sentiment or mood analysis work, I employ temporal dependencies between documents. Second, the work assumes continuous embedding while other supervised topic models assume a discrete set of labels (multiclass). Third, I use specific time information rather than only ordering between documents, which is covered in CRF.

3.3.2 Temporal Dynamics of Binary Sentiment

I augment the standard dataset in sentiment analysis $\{(X^{(i)}, Y^{(i)}), i = 1, \ldots, n\}$ with time stamps $T^{(i)} \in \mathbb{R}$, representing the time document $X^{(i)}$ was authored. I assume that the documents are represented as feature vectors $X^{(i)}$. The feature vector can have any document-level features such as bag-of-words or even sophisticated autoencoder features. The response variables $Y^{(i)} \in \{-1, +1\}$ are binary sentiment polarity values. I additionally assume a latent variable $Z^{(i)} \in \mathbb{R}$ associated with $X^{(i)}$ and $Y^{(i)}$

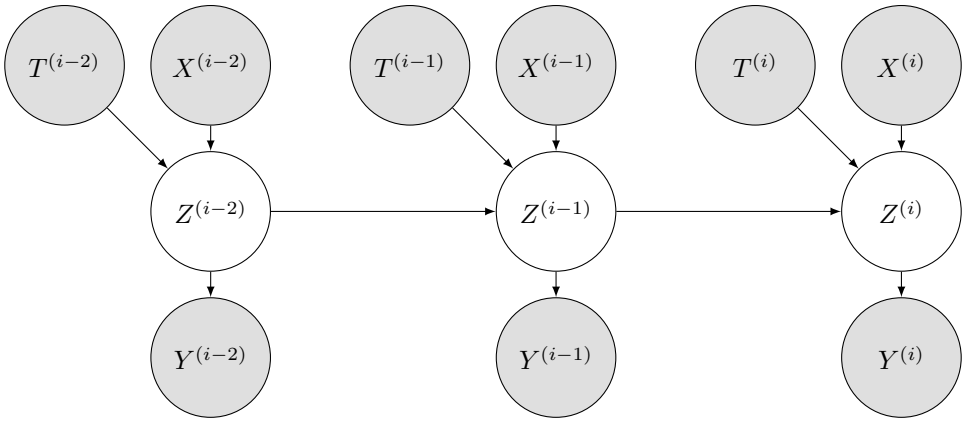

Figure 7: Graphical model of the temporal sentiment analysis model. X denote sequence of documents, T is the corresponding authoring time, Y is for the sentiment, and Z is the continuous latent variable. See text for details.

representing a continuous sentiment concept similar to Section 3.2.

The introduction of the continuous latent variable serves several roles: (i) it is easier to construct temporal models in continuous state space, and (ii) the framework conveniently generalizes to mood analysis where there are a large number of emotions embedded in a low dimensional continuous space (we explore this generalization in the next section).

I assume that $Z|Y$ follows a Gaussian distribution in $\mathbb{R}$, implying that Y is a (stochastic) discretization of Z. I also assume that $Z|X$ follows a linear regression model and that $Z^{(1)}, \ldots, Z^{(n)}$ follow a Markov chain with Gaussian conditional distributions $Z^{(i)}|Z^{(i-1)}$. The formal definition appears below and the graphical model appears in Figure 7.

1. $X^{(i)} \to Z^{(i)} \to Y^{(i)}$ forms a Markov chain. In other words, $Y^{(i)}$ is independent of $X^{(i)}$ given $Z^{(i)}$.

2. The distribution of $Z|Y$ is a Gaussian with appropriate mean and variance:

$$\{Z^{(i)}|Y^{(i)} = y\} \sim \mathcal{N}(y, \sigma_y^2),\ y \in \{-1, +1\}.$$

3. The distribution of $Z|X$ follows a linear regression model (assuming the document X is represented as a vector)

$$\{Z^{(i)}|X^{(i)} = x\} \sim \mathcal{N}(\theta^\top x, \epsilon^2),\ X^{(i)}, \theta \in \mathbb{R}^k. \tag{10}$$

4. The latent variables follow a Markov chain with Gaussian conditionals.

$$\{Z^{(i)}|Z^{(i-1)}\} \sim \mathcal{N}(Z^{(i-1)}, \beta\Delta T) \text{ where } \Delta T = T^{(i)} - T^{(i-1)} \tag{11}$$

Assumptions 3 and 4 above can be combined to produce

$$\begin{aligned}
&\{Z^{(i)}|X^{(i)},Z^{(i-1)}\} \sim \mathcal{N}(\mu^{(i)}, \sigma^{(i)}) \\
&\text{where } \sigma^{(i)} = \left((\beta\Delta T)^{-1} + \epsilon^{-2}\right)^{-1} \\
&\qquad \mu^{(i)} = \sigma^{(i)} \left((\beta\Delta T)^{-1} Z^{(i-1)} + \epsilon^{-2}\theta^\top X^{(i)}\right).
\end{aligned}$$

For simplicity, we consider (above and in the sequel) the time points $T^{(i)}$ to be non-random, and we therefore omit them in the probability notations for example $P(Z^{(i)}|X^{(i)})$ rather than $P(Z^{(i)}|X^{(i)}, T^{(i)})$. This is analogous to fixed design in regression analysis, as opposed to random design.

I emphasize the following characteristics. First, low-dimensional latent variable $Z^{(i)}$ solely determines polarity $Y^{(i)}$ ($Y^{(i)}$ is independent of high-dimensional $X^{(i)}$ given $Z^{(i)}$). Second, $Z^{(i)}$ has a distribution that centered at $Z^{(i-1)}$ with a variance that increases with ΔT, which is in agreement with psychological observations as well as standard models in the time series literature. Third, as we do not specify $p(X)$, the model is a discriminative model. It is similar to standard discriminative structured classifiers (such as CRF) with an additional constraint for inter-document dependency by β and ΔT. It matches our intuition as temporal proximity tends to imply proximity in sentiment (for blog posts written by the same author).

3.3.2.1 Learning and Using the Model

Parameters $\eta = (\theta, \beta, \mu_y, \sigma_y)$ can be estimated by maximizing the conditional likelihood, $p(Y|X)$, of observed data.

We consider two alternatives to handle documents written by different authors: (i) estimating a single set of parameters for all authors, and (ii) estimating separate parameters for each author. In the first approach, the model is universal and can capture generic trends. The second approach fits specialized models for each author. While the first approach appears more limited than the second, it is particularly useful when some authors do not have sufficient labeled data. In either case, we maximize the likelihood function for the observed data, which integrates over the latent variables.

In the first case above, we estimate the parameter by maximizing the total sum of conditional log-likelihood (temporal dependencies only holds for each author). Denoting the set of authors by A and an individual author as $a \in A$, we have

$$\eta = \arg\max_{\eta} \sum_{a \in A} \ell(\eta, a) \tag{12}$$

$$\begin{aligned} \ell(\eta, a) &= \log p_\eta(y_a^{(1)}, \ldots, y_a^{(n)} | x_a^{(1)}, \ldots, x_a^{(n)}) \\ &= \log \int_z p_\theta(z^{(1)} | x_a^{(1)}) \cdot \prod_{i=2}^{n} p_{\theta,\beta}(z^{(i)} | z^{(i-1)}, x_a^{(i)}) \cdot \prod_{i=1}^{n} p_{\mu_y,\sigma_y}(y_a^{(i)} | z^{(i)}) dz, \end{aligned} \tag{13}$$

where $x_a^{(i)}$ and $y_a^{(i)}$ denote documents and labels associated with a specific author $a \in A$ and the integral over z represents integration over the latent variables $z^{(i)}$. In the second case, the log-likelihood is the same as above except that we have multiple terms of log-likelihood for each author with different parameters $\eta_a, a \in A$. It can possibly describe behaviors of an author in depth (some authors may have stronger temporal dependency or lesser).

3.3.2.2 Inference

To predict the most likely polarity of a given temporal document (in test time), we compute below the most likely value. Note that we use the previous time stamped

documents to improve the estimation accuracy.

$$\hat{y}^{(i)} = \arg\max_{y^{(i)}} p(y^{(i)}|x^{(i)}, \ldots, x^{(1)}) \tag{14}$$

$$= \arg\max_{y^{(i)}} \int \cdots \int p(y^{(i)}|z^{(i)}) \cdot \prod_{j=1}^{i} p(z^{(j)}|x^{(j)}, z^{(j-1)}) dz^{(1)} \ldots dz^{(i)} \tag{15}$$

3.3.2.3 Approximation and Implementation

There are several approaches for computing (13) including numeric integration, Laplace approximation, and Markov Chain Monte Carlo (MCMC). I use a simpler approximation that replaces the Gaussian distribution over Z with a Dirac's delta centered at the most likely value of z as I did in Section 3.2.2.2. This approximation replaces the integral with the integrand, evaluated at the most likely value of the latent variable. Naturally, the approximation quality increases as the variance of the Gaussian decreases.

$$\int \mathcal{N}(z\,; z^*, \sigma) g(z) dz \approx c(\sigma) \int \delta(z - z^*) g(z) dz = c(\sigma) g(z^*).$$

Applying this approximation to (13) we get:

$$\begin{aligned} \ell &\approx \log p(y^{(1)}, \ldots, y^{(n)}, z^{(1)} = z^{*(1)}, \ldots, z^{(n)} = z^{*(n)} | x^{(1)}, \ldots, x^{(n)}) \\ &= \sum_{i=1}^{n} \log p_\theta(z^{*(1)}|x^{(1)}) + \sum_{i=2}^{n} \log p_{\theta,\beta}(z^{*(i)}|z^{*(i-1)}, x^{(i)}) + \sum_{i=1}^{n} \log p_{\mu_y,\sigma_y}(y^{(i)}|z^{*(i)}) \end{aligned} \tag{16}$$

where $z^{*(i)} = \arg\max_z p(z^{*(i)}|z^{*(i-1)}, x^{(i)}) p(y^{(i)}|z^{*(i)})$

$$= \left({\sigma^{(i)}}^{-1} + {\sigma_y}^{-1}\right)^{-1} \left({\sigma^{(i)}}^{-1} \mu^{(i)} + {\sigma_y}^{-1} \mu_y\right).$$

The maximum likelihood estimator for θ, $\hat{\theta} = \arg\max_\theta$

$$\left[\log p(z^{*(1)}|x^{(1)}) + \sum_{i=2}^{n} \log p(z^{*(i)}|z^{*(i-1)}, x^{(i)})\right]$$

is conveniently given in closed form. Since the conditional probabilities are Gaussian, the MLE is equivalent to a least squares linear regression model $\theta^\top X = W$ where

Table 3: Polarity assignments of moods

±	**Assignments of moods**
+1	excited, pleased, good, cheerful, amused, hopeful, bouncy, chipper, thoughtful, accomplished
-1	cold, exhausted, sleepy, tired, bored, sick, sore, uncomfortable, depressed, sad, annoyed

$X = [x^{(1)}, \ldots, x^{(n)}]$ and $W \in \mathbb{R}^n$ is

$$W = \begin{bmatrix} z^{*(1)} \\ \vdots \\ \epsilon^2 \Big[\big((\beta \Delta T)^{-1} + \epsilon^{-2} \big) z^{*(i)} - (\beta \Delta T)^{-1} z^{*(i-1)} \Big] \\ \vdots \end{bmatrix}.$$

After obtaining the MLE $\hat{\theta}$, we can fix that parameter value and compute the MLE for the remaining parameters $(\hat{\beta}, Z^*, \hat{\mu}_y, \hat{\sigma}_y)$ using standard gradient based optimization. We then re-calculate $\hat{\theta}$ and iterate until convergence.

The approximation above can also be used in test time prediction. It proceeds as above by replacing the integrals over the latent variables Z by the integrand evaluated at the most likely value of the latent variables.

3.3.2.4 Experiment

Dataset Most standard sentiment analysis datasets [62, 85, 18] focus on a sentiment concept corresponding to opinions or reviews on specific topics. This sentiment concept is unlikely to vary significantly with time as much because it reflect the author's opinion about a specific issue (however, see [28] that discovers some temporal effects in movie reviews). I choose instead to model blog posts, which depend more significantly on time.

I use similar dataset from Section 3.2.3.1. I gathered data by crawling a popular

blog service, Livejournal[4] from May 2010 to July 2011. Livejournal provides timestamps as well as emotion annotation that reflect the author's mood of a single blog post (one annotation per one blog post). The authors are offered the use of a wide range of emoticons and also offers a free text emotion annotation. The crawling resulted in two million documents authored by 315K authors, and about 20% of the authors annotated their posts with such annotation.

Since most authors do not provide more than 2 documents (the median number of blog post by an author), I selected the top 50 most frequently publishing authors. I also had to remove spam authors who kept posting the same content repeatedly with random emotions. I finally obtained 19 authors with 64 different emotions after removing rare emotions that appeared in less than 20 documents.

I want to note that most authors don't write frequently, which makes estimating their emotion harder. For example, the 50th most active author had only 52 documents for training. It is a common case when we handle social media data. I expect my method that exploits temporal dependencies to lead to a more accurate model than non-temporal models in this sparse setting.

In this section, I converted the emotions to polarity labels $\{-1, +1\}$, while ignoring neutral emotions as indicated by Table 3. This procedure yielded 6,295 documents with 0.27:0.73 (negative:positive) class distributions. Section 3.3.3.1 describes experiments on the entire set of emotions including the neutral emotions.

For document-level features, $X^{(i)}$, I selected negation-handled term frequencies. Specifically, I tokenized and stemmed (Krovetz) the documents, and added new features for negated words as is commonly done in sentiment analysis [26]. For example, this implies that negation words like "not" or "isn't" followed by "good" will yield negated tokens such as "not-good." This procedure yielded 43,910 features. To create

[4] `http://www.livejournal.com`

Table 4: Test set F1 and accuracy results for predicting sentiment polarity and corresponding training time of each method. Bold face shows statistically significant improvement over other competitors (t-test, 95% confidence).

Methods	**F1**	**Accuracy**	**Time(sec)**
temporal sentiment method	**0.7596**	**0.8058**	16.96
temporal linear-chain CRF	0.7130	0.7742	3352.81
temporal VARX(1)	0.6554	0.7172	9.17
non-temporal logistic regression	0.7109	0.8016	48.39
non-temporal SVM	0.6555	0.7557	27.26
non-temporal SLDA	0.5093	0.7379	1047.59
non-temporal naive Bayes	0.6915	0.7373	1.07

the final document representation, I used the square root of the normalized term frequency of the tokens since (i) it is closely associated with the Hellinger distance and the multinomial Fisher geometry that have nice theoretical properties [10], and (ii) it often leads to improved modeling accuracy (see [30, 35, 37, 36, 15] for examples of using Hellinger distances in text modeling and interpreting it in terms of information geometry).

It is possible other sophisticated features such as dependency tree features or autoencoder features can be used for $X^{(i)}$; however, the term frequency feature is sufficient for demonstrating the model and contrasting it with standard sentiment analysis models. The main contribution is presenting the manifold Z which is applicable regardless of the document-level feature extraction.

A few statistics concerning the dataset are: (i) the average document length is 71 words ($\pm$89), (ii) most documents (55%) have less than 50 words, (iii) there are a few very long documents (0.01%) with more than 500 words, and (iv) the average word length is 8.33 characters. This denotes the blog posts are relatively sparser than other popular sentiment dataset such as the movie review dataset [62].

Classification One of the primary tasks in sentiment analysis is predicting the sentiment polarity $Y \in \{-1, +1\}$ of a document X. Table 4 compares seven different methods in sentiment prediction with my model using shared parameters across all authors (Equation 12). With larger dataset, we may use author-specific parameters to improve the accuracy.

I compare the temporal sentiment method with non-temporal classification baselines including a well known supervised topic model (Supervised Latent Dirichlet Allocation[5] [6]), SVM[6], logistic regression with L_2 regularization, and naive Bayes classifier. I also compare my model with temporal models such as Conditional Random Field[7] [31] and VARX(1)[8], $y^{(t)} = a + X^{(t)}\beta + Ay^{(t-1)}$, which is the most relevant vector autoregressive model. The inputs of VARX(1) model are given $\{-1, +1\}$ accordingly and the output is interpreted as a binary classification by its polarity.

Training and test set were split 50:50 by post-by-post random (not author-based). Post orderings of each author were recovered using timestamps $T^{(i)}$ and author ID after the split. All regularization parameters were chosen by a validation set on training examples using grid search of 5 candidates. I tried various sizes of latent topics on SLDA (2 to 20) and included the best result.

The model demonstrates statistically significant improvement in both F1 measure and classification accuracy (t-test with 100 random trials) compared to all non-temporal methods and other temporal methods. SLDA performed relatively poorly especially on F1, perhaps due to the extreme sparsity. CRF and VARX(1) do not explicitly model the time gap (ΔT) between observations and rather only consider the ordering of the time stamps unlike the proposed model. This explains the lower performance of CRF, which is generally considered a top performing model.

[5]Gibbs sampling implementation: `https://github.com/michaelchughes/SuperTopicModels`

[6]LibSVM (ν-SVM, RBF kernel): `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[7]Linear-chain CRF from Kevin Murphy: `http://www.cs.ubc.ca/~murphyk/Software/CRF/crf.html`

[8]Matlab implementation

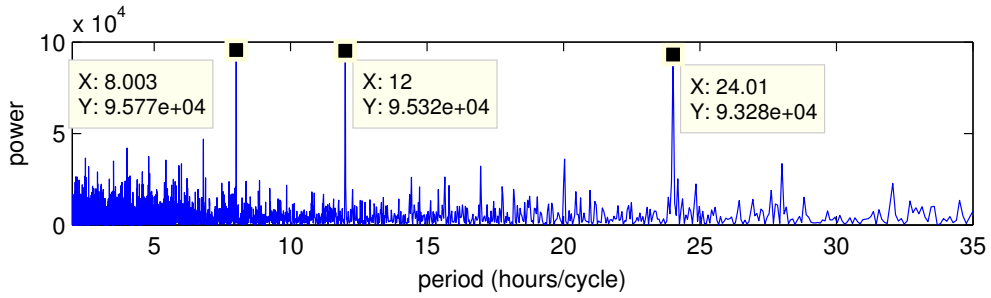

Figure 8: Fourier components of the latent variable in the global model. There are three significant periodic components representing the periods: 8 hours, 12 hours, and 24 hours (circadian rhythm).

The rightmost column of Table 4 shows average training time of each methods. My method (temporal sentiment) takes much shorter time to train the model compared to methods that have comparable performance (such as CRF, SVM or logistic regression). In fact, it is far faster than CRF and SLDA, which are considered to be one of the state-of-the-art methods.

Periodicity The temporal model can also be used for analyzing temporal variations in the emotions of authors. Figure 8 investigates the periodic behavior of the temporal sentiment by displaying the Fourier components of the latent variable. The x axis of the graph measures period (one over frequency) rather than traditional Fourier frequency for better interpretation in this context.

The peaks in graph show significant periodic components at 8 hours, 12 hours, and 24 hours. Interestingly, the strong 24 hours periodicity matches the circadian rhythm discovery from chronobiology and psychology research [57, 20]. Specifically, this confirms the work of [57] that explored a circadian component in positive affect. The confirmation is noteworthy as my model was constructed from blog posts, while they surveyed human subjects in a controlled environment.

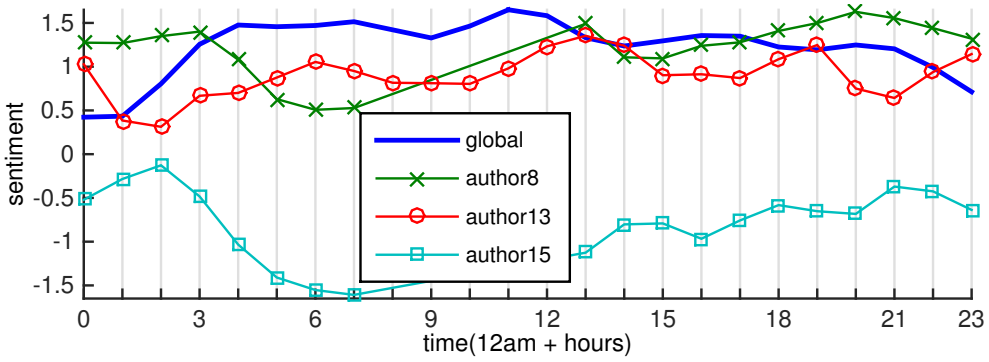

Figure 9: Hourly pattern of global sentiment and selected authors. The y axis correspond to the latent variable of the model (higher values correspond to stronger positive sentiment). See text for details.

Hourly Pattern From Figure 8, we found 24 hours of periodicity. We investigate this finding further by visualizing hourly average sentiment values (as measured by the latent variable) for models trained on data of specific authors and for the globally-trained model (Figure 9).

The figure shows interesting observations that agree our intuition. First, the trend of global model is well aligned with generic daily schedules. Four local maxima at 6am, 11am, 16pm and 20pm match to positive daily events: wake-up time, lunch break, office closing time, and dinner time. There are notable minima in late night (0am-3am) and the end of lunch break (1pm). The highest and lowest sentiment values are achieved around noon and midnight, respectively. Second, some authors have different temporal patterns, which may indicate different time zones or different life styles. For example, author 8 and 15 exhibit low sentiment in the early morning and high sentiment late at night. Third, the hourly trend shows characteristics of an author. A sentiment structure of author 15 shows an overall lower sentiment compared to others. When I manually visit the actual blog site of the author, I observed many negative annotations. These observations can be useful in psychological studies as well as marketing and advertisement sciences.

3.3.3 Temporal Dynamics of Multivariate Emotions

I now extend the temporal dynamics model to the case where a richer concept describing a diverse set of emotions rather than having a one dimensional polarity. Such emotions, for example `happy`, `sad`, `excited`, and `tired` are correlated with the polarity, but they offer an opportunity to construct a more fine grained model of the author's emotion.

Since there is a relatively large set of possible emotions, and these emotions are related to one another, I avoid constructing a separate temporal dynamics model for each individual binary emotion. Instead, I extend the formalism in Section 3.2, where a large set of emotions are embedded in a low dimensional Euclidean space. I thus generalize the model of the previous section by increasing the dimensionality of the latent variable Z. I make use of the same notation as Section 3.3.2, but now $Y^{(i)} \in \{1, 2, \ldots, |C|\}$, and the latent variable is a vector $Z^{(i)} \in \mathbb{R}^l$.

The largest difference with the one in Section 3.2 is that it lacks temporal dependence. Section 3.2 assume blog posts are independent from each other while this model makes better use of data considering previous observations. Another difference is that I consider both individual-level and global-level emotion manifolds, as described in the previous section.

The assumptions in Section 3.3.2 are now extended to adapting multivariate emotions. Assumption 2 now has a multivariate Gaussian instead of the univariate Gaussian: $\{Z^{(i)}|Y^{(i)} = y\} \sim \mathcal{N}(\mu_y, \Sigma_y)$; the centroids (μ_y) correspond to $\{-1, +1\}$ from the previous section that described positive and negative polarity in the latent space. Assumption 3 is extended to multi-response regression: $\{Z^{(i)}|X^{(i)} = x\} \sim \mathcal{N}(\Theta x, \epsilon^2 \, \mathsf{I}), \Theta \in \mathbb{R}^{l \times k}$, and a spherical covariance is used instead of a scalar in Assumption 4: $\{Z^{(i)}|Z^{(i-1)}\} \sim \mathcal{N}(Z^{(i-1)}, \beta \cdot \Delta T \cdot \mathsf{I})$.

I additionally introduce the fifth and last assumption similar to that in Section 3.2.

5. $\forall y \in C$, distances between $\left\{\mathsf{E}[Z^{(i)}|Y^{(i)} = y]\right\}$ are similar to the corresponding

distances in $\{\mathsf{E}[X^{(i)}|Y^{(i)} = y]\}$.

This assumption enables us to estimate μ_y, Σ_y by multi-dimensional scaling (MDS) on empirical averages corresponding to $\mathsf{E}[X|Y = y]$.

It is worth mentioning this model incorporates two different types of proximities: temporal proximities between $Z^{(i)}$ and $Z^{(i-1)}$ and spatial proximities between $\mathsf{E}[Z^{(i)}|Y^{(i)}]$ and $\mathsf{E}[X^{(i)}|Y^{(i)}]$. I also expect the dimensionality l of the latent space $Z^{(i)}$ to be smaller than the number of emotions $l \ll |C|$ and much smaller than the dimensionality of $X^{(i)}$. This means that $Z^{(i)}$ serves as a latent low-dimensional variable that connects two observed high dimensional variables.

To estimate the model, we start by MDS to estimate μ_y, Σ_y. The dimension of latent space Z is bounded by $|C| - 1$ consequently. We then follow the maximum likelihood procedure and approximation as described previously.

3.3.3.1 Experiments

Dataset We now consider full emotion labels (including all neutral emotions) from Section 3.3.2.4. The data contains 11,659 documents with 43,910 features and 64 emotions. The label frequencies varied between 0.0018 (21 documents) to 0.1441 (1680 documents), with an average label frequency of 0.0156 (182 documents).

Hourly Pattern Figure 10 shows temporal trajectories of the latent variable in several models based on individuals, as well as a global model on the first two dimensions of the model. The first two dimensions seem to capture sentiment level (horizontally) and energy level (vertically). Gray words show centroids of corresponding emotion labels.

The global model shows a progression from negative sentiment (left side) to positive sentiment (right side) from 12am, and the progression is reversed later on 12pm similar to Figure 9. Each author shows their unique progression style and location. For example, author 15 is exceptional in its location (close to negative emotions) and

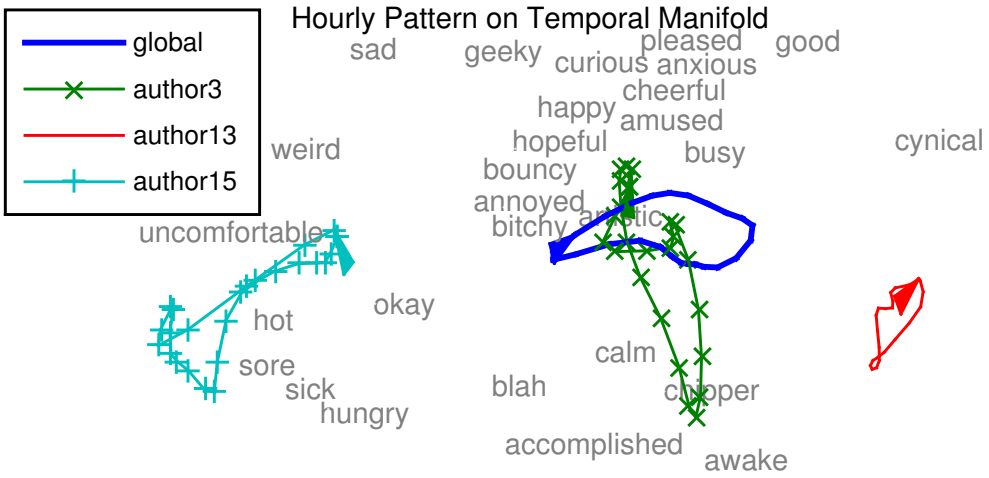

Figure 10: Hourly trends of global model and selected authors on the first two dimensions of the manifold (smoothed). Gray words show $\mathsf{E}[Z|Y = y]$. The arrow shows the start of the day (12am) and direction of the progression of each circle. There is clear separation between day and night time.

its progression (counter-clockwise) as opposed to the frequently observed clockwise progression; this means the author has emotional transitions in an opposite order to most people.

Classifying Emotions We now consider the task of predicting the emotion of a given text document. Compared to classifying a binary sentiment polarity, predicting the multiclass emotion concept is difficult due to the large number of interrelated

Table 5: Test set F1 and accuracy results for predicting emotion among 64 emotions. Bold face shows statistically significant improvement over other competitors (t-test, 95% confidence). See text for details.

	Macro F1	**Accuracy**
temporal model (proposed)	**0.2552**	0.4381
non-temporal model	0.2522	0.4370
logistic regression	0.1618	0.4329
SLDA	0.1131	0.3358
naive Bayes	0.0103	0.1405

emotions, some of which having only a small set of labeled documents.

Table 5 shows the test set classification performance of predicting the emotion (out of a set of $|C| = 64$ possible emotions). I compared the temporal model with global parameter setting, the non-temporal model (Section 3.2), SLDA [6], logistic regression, and naive Bayes. CRF failed to converge in weeks of running time. Details of experimental configuration remain the same as the temporal binary sentiment experiment (Section 3.3.2.4) except that (i) the size of latent topics on SLDA was varied from $2 \cdot |C|$ to $5 \cdot |C|$ and (ii) the experiment was repeated 50 times by random split.

The improvement in classification accuracy is noticeable, but not statistically significant; however, the improvement in macro F1 is statistically significant. This is notable as the class distribution is highly unbalanced and F1 is the measure that is often regarded as more informative than accuracy in the multiclass case with unbalanced class frequencies.

Above, the dimension of the latent space l is $|C| - 1$. However, reducing the dimensionality to $0.7 \cdot |C|$, $0.45 \cdot |C|$, and $0.19 \cdot |C|$ reduces the accuracy of the full model only to 90%, 80%, and 70% respectively. This shows that the manifold includes most of its information in only a few dimensions. This classification experiment includes tiny classes that have a few documents (minimum 20 documents). When we performed the same experiment while excluding tiny classes (minimum 100 documents), the non-temporal method (Section 3.2) performed better in terms of emotion classification accuracy. I conclude that my model is well suited to handling small classes, which is often the case in self reported mood data.

3.3.4 Summary and Discussion

I presented a temporal statistical model, which extends the model in Section 3.2, for modeling binary sentiment polarities and multivariate emotions. The model uses

temporally-dependent continuous latent variables in order to capture relationships between various emotions and observations. I examined a wide variety of applications, including sentiment or emotion prediction of time-stamped documents and scientific research into the temporal variations of human emotions. The experimental results demonstrate improved statistical modeling and confirm discoveries from the psychological literature concerning the static and temporal structure of human emotions.

The model in this section has improved the previous model (Section 3.2) in various aspects of the *good representation* criteria (Section 1.4).

1. Reconstruction Quality: Compared to the non-temporal model, the temporal model maintains temporal dependencies between documents in order to preserve additional sequential nature of the original data.

2. Discriminative Power: The temporal model has improved classification results compared to the one without the temporal dynamics. Experimental results are in Section 3.3.2.4 and Section 3.3.3.1.

3. Interpretability: In additional to presenting an accessible emotion manifold similar to the one in Section 3.2, the temporal model captures temporal characteristics of human emotions such as a circadian rhythm and hourly patterns (Figure 8,9,10).

4. Computation: Similar to the previous model, we can efficiently compute the representation based on Dirac's delta approximation.

3.4 Chapter Discussion

In this chapter, we have discussed new ways to estimate document representations utilizing label characteristics that reveal useful information of document space. The new representations presented in this chapter improved the evaluation metric of the *good representation* (Section 1.4) in several ways.

Although this chapter solely focused on emotion prediction problems, the framework can be extended to various domains. The framework can be directly applied when we have labels with latent structures. Categorizing genre of a literature or a music are good examples because genre also has a continuous structural characteristic. Another example is multi-label prediction problems since they often involve non-exclusive relationships of labels.

In the next chapter, I will focus on another challenge in document representation learning: the sequential nature of a document. Instead of sitting with the standard bag-of-words features, I will explore much richer features employing sequential information. In Chapter 5, I will consider a unified view of utilizing both labels and sequential information.

CHAPTER IV

EMPLOYING SEQUENTIAL INFORMATION

4.1 Sequential Information

Various levels of sequentiality are frequently observed during a document modeling and play critical role in the document's semantics. While word-level sequentiality modifies the meaning of phrases and sentences, sentence-level or paragraph-level sequentiality alters the organization of a document. Inter-document sequentiality helps us to understand a discourse and document revisions reveal the development process of a document.

A document modeling with various levels of sequential dependencies is difficult for two reasons. First, a large portion of dependencies are not visible and are needed to be uncovered. There is no simple way for the extraction. Numerous natural language processing researchers are solving this problem in grammar parsing, dependency parsing, and coreference resolution. Second, since textual data has been already sparse, introducing another layer of contextual dependencies makes the data even sparser. Multiple observations of a single term need to be treated differently in different contexts, which drastically adds up dimensionality of the data.

This section focuses on the second type of difficulty while employing readily apparent sequentiality such as word-level sequentiality and revision histories. There have been various attempts to model those evident sequentiality especially on word-level. For example, n-gram model is the most famous attempt to capture local word-level dependencies although it suffers heavily from the sparsity issue. See Section 2.2 more discussions.

Approaches in this chapter differ from related studies mostly in their perspective

of a document. Most other work models a document as a series of inter-related observations. Unlike this traditional viewpoint, my proposed approaches formulate a joint or conditional distribution between a word and its positional information, $p(w, t)$ or $p(w|t)$. This perspective enables flexible extensions over different levels of sequentiality as well as more a compact way to represent local dependencies.

Section 4.2 discusses a new document representation employing two levels of sequential information, spatial and temporal, in order to model a version-controlled document. Section 4.3 presents a new way to capture local word dependencies preventing negative effects of sparsity.

4.2 Spatial and Temporal Sequential Information

4.2.1 Modeling Version-controlled Documents

Most computational linguistics studies concentrate on modeling or analyzing documents as sequences of words. In section, we consider modeling and visualizing version-controlled documents which is the authoring process leading to the final word sequence. In particular, I focus on documents whose authoring process naturally segments into consecutive versions. The revisions, as the differences between consecutive versions are often called, may be authored by a single author or by multiple authors working collaboratively.

One popular way to keep track of version-controlled documents is using a version control system such as CVS, Subversion (SVN), or GIT. This is often the case with books or with large computer code projects. In other cases, more specialized computational infrastructure may be available, as is the case with the authoring API of Wikipedia.org, Slashdot.com, and Google Wave. Accessing such API provides information about what each revision contains, when was it submitted, and who edited it. In any case, we formally consider a version-controlled document as a sequence of documents $d_1, \ldots, d_l$ indexed by their revision number where d_i typically contains

some locally concentrated additions or deletions, as compared to d_{i-1}.

In this work, I develop a continuous representation of version-controlled documents that generalizes the locally weighted bag of words representation [39]. The representation smooths the sequence of version-controlled documents across two axes-space s and time t. The space axis s represents document position and the time axis t represents the revision. The smoothing results in a continuous map from a space-time domain to the simplex of term frequency vectors

$$\gamma : \Omega \to \mathbb{P}_V \quad \text{where} \quad \Omega \subset \mathbb{R}^2, \quad \text{and} \tag{17}$$
$$\mathbb{P}_V = \left\{ w \in \mathbb{R}^{|V|} : w_i \geq 0, \ \sum_{i=1}^{|V|} w_i = 1 \right\}.$$

The mapping above (V is the vocabulary) captures the variation in the local distribution of word content across time and space. Thus $[\gamma(s,t)]_w$ is the (smoothed) probability of observing word w in space s (document position) and time t (version), $p(w,s,t)$. Geometrically, γ realizes a divergence-free vector field (since $\sum_w [\gamma(s,t)]_w = 1$, γ has zero divergence) over the space-time domain Ω.

We consider the following four version-controlled document analysis tasks. The first task is visualizing word-content changes with respect to space (how quickly the document changes its content), time (how much does the current version differs from the previous one), or mixed space-time. The second task is detecting sharp transitions or edges in word content. The third task is concerned with segmenting the space-time domain into a finite partition reflecting word content. The fourth task is predicting future revisions. The main tool in addressing tasks 1-4 above is to analyze the values of the vector field γ and its first order derivatives fields

$$\nabla\gamma = (\dot{\gamma}_s, \dot{\gamma}_t)\,. \tag{18}$$

4.2.2 Space-Time Smoothing for version-controlled documents

With no loss of generality, we identify the vocabulary V with positive integers $\{1, \ldots, V\}$ and represent a word $w \in V$ by a unit vector[1] (all zero except for 1 at the w-component)

$$e(w) = (0, \ldots, 0, 1, 0, \ldots, 0)^\top \quad w \in V. \tag{19}$$

I extend this definition to word sequences thus representing documents $\langle w_1, \ldots, w_N \rangle$ ($w_i \in V$) as sequences of V-dimensional vectors $\langle e(w_1), \ldots, e(w_N) \rangle$. Similarly, a version-controlled document is sequence of documents $d^{(1)}, \ldots, d^{(l)}$ of potentially different lengths $d^{(j)} = \langle w_1^{(j)}, \ldots, w_{N(j)}^{(j)} \rangle$. Using (19), I represent a version-controlled document as the array

$$\begin{array}{ccc} e(w_1^{(1)}), & \ldots, & e(w_{N(1)}^{(1)}) \\ \vdots & \ddots & \vdots \\ e(w_1^{(l)}), & \ldots, & e(w_{N(l)}^{(l)}) \end{array} \tag{20}$$

where columns and rows correspond to space (document position) and time (versions).

The array (20) of high dimensional vectors represents the version-controlled document without any loss of information. Nevertheless the high dimensionality of V suggests that we smooth the vectors in (20) with neighboring vectors in order to better capture the local word content. Specifically, I convolve each component of (20) with a 2-D smoothing kernel K_h to obtain a smooth vector field γ over space-time [79] e.g.,

$$\gamma(s,t) = \sum_{s'} \sum_{t'} K_h(s - s', t - t') e(w_{s'}^{(t')})$$

$$K_h(x,y) \propto \exp\left(-(x^2 + y^2)/(2h^2)\right). \tag{21}$$

[1]Note the slight abuse of notation as V represents both a set of words and an integer $V = \{1, \ldots, V\}$ with $V = |V|$.

Thus as (s,t) vary over a continuous domain $\Omega \subset \mathbb{R}^2$, $\gamma(s,t)$, which is a weighted combination of neighboring unit vectors, traces a continuous surface in $\mathbb{P}_V \subset \mathbb{R}^V$. Assuming that the kernel K_h is a normalized density it can be shown that $\gamma(s,t)$ is a non-negative normalized vector i.e., $\gamma(s,t) \in \mathbb{P}_V$ (see (17) for a definition of $\mathbb{P}_V$) measuring the local distribution of words around the space-time location (s,t). It thus extends the concept of lowbow (locally weighted bag of words) introduced in [39] from single documents to version-controlled documents.

One difficulty with the above scheme is that the document versions $d_1, \ldots, d_l$ may be of different lengths. We consider two ways to resolve this issue. The first pads shorter document versions with zero vectors as needed. I refer to the resulting representation γ as the non-normalized representation. The second approach normalizes all document versions to a common length, say $\prod_{j=1}^{l} N(j)$. That is each word in the first document is expanded into $\prod_{j \neq 1} N(j)$ words, each word in the second document is expanded into $\prod_{j \neq 2} N(j)$ words etc. I refer to the resulting representation γ as the normalized representation.

The non-normalized representation has the advantage of conveying absolute lengths. For example, it makes it possible to track how different portions of the document grow or shrink (in terms of number of words) with the version number. The normalized representation has the advantage of conveying lengths relative to the document length. For example, it makes it possible to track how different portions of the document grow or shrink with the version number relative to the total document length. In either case, the space-time domain Ω on which γ is defined (21) is a two dimensional rectangular domain $\Omega = [0, I] \times [0, J]$.

Before proceeding to examine how γ may be used in the four tasks described previously, I demonstrate my framework with a simple low dimensional example. Assuming a vocabulary of two words $V = \{1, 2\}$, I can visualize γ by displaying its first

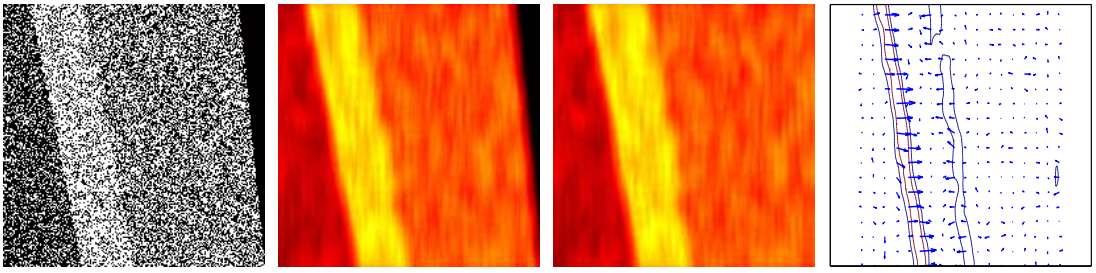

Figure 11: Four space-time representations of a simple synthetic version-controlled document over $V = \{1, 2\}$ (see text for more details). The left panel displays the first component of (20) (non-smoothed array of unit vectors corresponding to words). The second and third panels display $[\gamma(s,t)]_1$ for the non-normalized and normalized representations respectively. The fourth panel displays the gradient vector field $(\dot{\gamma}_s(s,t), \dot{\gamma}_t(s,t))$ (contour levels represent the gradient magnitude). The black portions of the first two panels correspond to zero padding due to unequal lengths of the different versions.

component as a grayscale image (since $[\gamma(s,t)]_2 = 1 - [\gamma(s,t)]_1$ the second component is redundant). Specifically, I created a version-controlled document with three contiguous segments whose $\{1, 2\}$ words were sampled from Bernoulli distributions with parameters 0.3 (first segment), 0.7 (second segment), and 0.5 (third segment). That is, the probability of getting 1 is highest for the second segment, equal for the third and lowest for the first segment. The initial lengths of the segments were 30, 40 and 120 words with the first segment increasing and the third segment decreasing at half the rate of the first segment with each revision. The length of the second segment was constant across the different versions. Figure 11 displays the non-smoothed ragged array (20) (left), the non-normalized $[\gamma(s,t)]_1$ (middle left) and the normalized $[\gamma(s,t)]_1$ (middle right).

While the left panel doesn't distinguish much between the second and third segment the two smoothed representations display a nice segmentation of the space-time domain into three segments, each with roughly uniform values. The non-normalized representation (middle left) makes it easy to see that the total length of the version-controlled document is increasing but it is not easy to judge what happens to the

relative sizes of the three segments. The normalized representation (middle right) makes it easy to see that the first segment increases in size, the second is constant, and the third decreases in size. It is also possible to notice that the growth rate of the first segment is higher than the decay rate of the third.

4.2.3 Visualizing Change in Space-Time

I apply the space-time representation to four tasks. The first task, visualizing change, is described in this section. The remaining three tasks are described in the next three sections.

The space-time domain Ω represents the union of all document versions and all document positions. Some parts of Ω are more homogeneous and some are less in terms of their local word distribution. Locations in Ω where the local word distribution substantially diverges from its neighbors correspond to sharp content transitions. On the other hand, locations whose word distribution is more or less constant correspond to slow content variation.

We distinguish between three different types of changes. The first occurs when the word content changes substantially between neighboring document positions within a certain document version. As an example consider a document location whose content shifts from high level introductory motivation to a detailed technical description. Such change is represented by

$$\|\dot{\gamma}_s(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial [\gamma(s,t)]_w}{\partial s} \right)^2 . \tag{22}$$

A second type of change occurs when a certain document position undergoes substantial change in local word distribution across neighboring versions. An example is erroneous content in one version being heavily revised in the next version. Such change along the time axis corresponds to the magnitude of

$$\|\dot{\gamma}_t(s,t)\|^2 = \sum_{w=1}^{V} \left(\frac{\partial [\gamma(s,t)]_w}{\partial t} \right)^2 . \tag{23}$$

Expression (22) may be used to measure the instantaneous rate of change in the local word distribution. Alternatively, integrating (22) provides a global measure of change

$$h(s) = \int \|\dot{\gamma}_s(s,t)\|^2 \, dt, \quad g(t) = \int \|\dot{\gamma}_t(s,t)\|^2 \, ds$$

with $h(s)$ describing the total amount of spatial change across all revisions and $g(t)$ describing the total amount of version change across different document positions. $h(s)$ may be used to detect document regions undergoing repeated substantial content revisions and $g(t)$ may be used to detect revisions in which substantial content has been modified across the entire document.

We conclude with the integrated directional derivative

$$\int_0^1 \|\dot{\alpha}_s(r)\dot{\gamma}_s(\alpha(r)) + \dot{\alpha}_t(r)\dot{\gamma}_t(\alpha(r))\|^2 \, dr \tag{24}$$

where $\alpha : [0,1] \to \Omega$ is a parameterized curve in the space-time and $\dot{\alpha}$ its tangent vector. Expression (24) may be used to measure change along a dynamically moving document anchor such as the boundary between two book chapters. The space coordinate of such anchor shifts with the version number (due to the addition and removal of content across versions) and so integrating the gradient across one of the two axis as in (23) is not appropriate. Defining $\alpha(r)$ to be a parameterized curve in space-time realizing the anchor positions $(s,t) \in \Omega$ across multiple revisions, (24) measures the amount of change at the anchor point.

The right panel of Figure 11 shows the gradient vector field corresponding to the synthetic version-controlled document described in the previous section. As expected, it tends to be orthogonal to the segment boundaries. Its magnitude is displayed by the contour lines which show highest magnitudes around segment boundaries.

Figure 12 shows the norm $\|\dot{\gamma}_s(s,t)\|^2$ (left), $\|\dot{\gamma}_t(s,t)\|^2$ (middle left) and the local maxima of $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|^2$ (middle right) for a portion of the version controlled Wikipedia Religion article. The first panel shows the amount of change in local

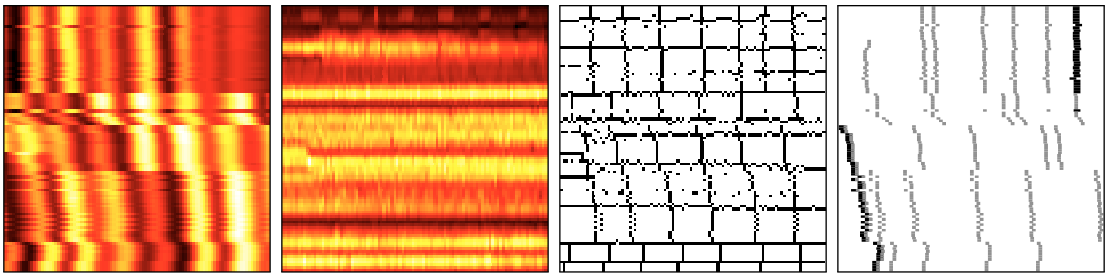

Figure 12: Gradient and edges for a portion of the version controlled Wikipedia Religion article. The left panel displays $\|\dot{\gamma}_s(s,t)\|^2$ (amount of change across document locations for different versions). The second panel displays $\|\dot{\gamma}_t(s,t)\|^2$ (amount of change across versions for different document positions). The third panel displays the local maxima of $\|\dot{\gamma}_s(s,t)\|^2+\|\dot{\gamma}_t(s,t)\|^2$ which correspond to potential edges, either vertical lines (section and subsection boundaries) or horizontal lines (between substantial revisions). The fourth panel displays boundaries of sections and subsections as black and gray lines respectively.

word distribution within documents. High values correspond to boundaries between sections, topics or other document segments. The second panel shows the amount of change as one version is replaced with another. It shows which revisions change the word distributions substantially and which result in a relatively minor change. The third panel shows only the local maxima which correspond to edges between topics or segments (vertical lines) or revisions (horizontal lines).

4.2.4 Edge Detection

In many cases documents may be divided to semantically coherent segments. Examples of text segments include individual news stories in streaming broadcast news transcription, sections in article or books, and individual messages in a discussion board or an email trail. For non-version-controlled documents finding the text segments is equivalent to finding the boundaries or edges between consecutive segments. See [21, 1, 51] for several recent studies in this area.

Things get a bit more complicated in the case of version-controlled documents. Segments, and their boundaries exist in each version. As in case of image processing,

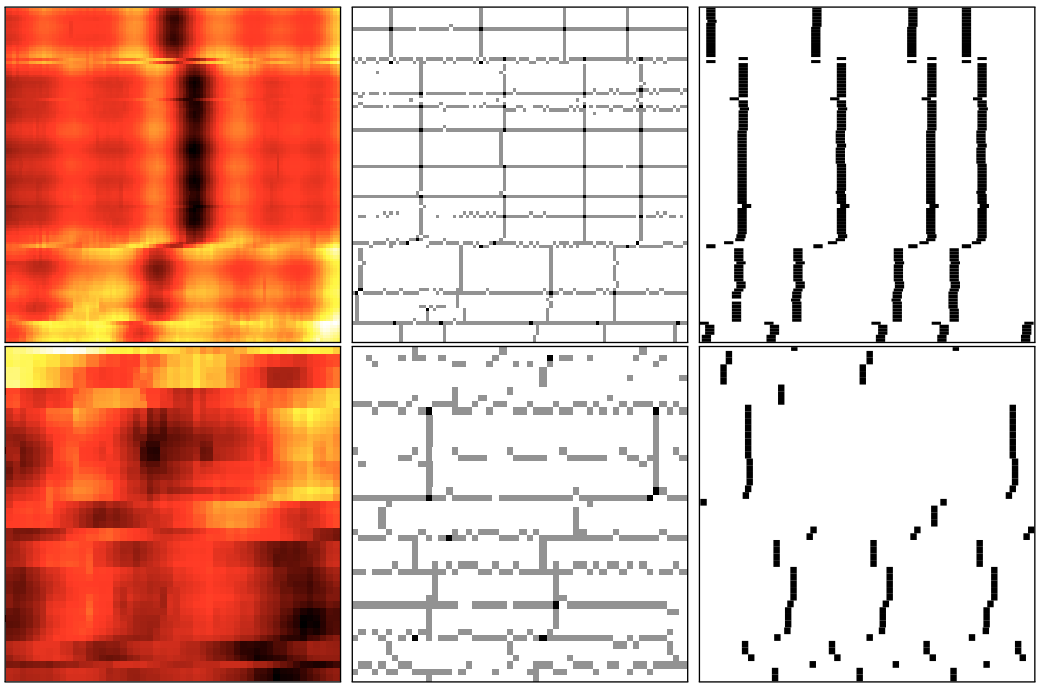

Figure 13: Gradient and edges of a portion of the version controlled Atlanta Wikipedia article (top row) and the Google Wave Amazon Kindle FAQ (bottom row). The left column displays the magnitude of the gradient in both space and time $\|\dot{\gamma}_s(s,t)\|^2 + \|\dot{\gamma}_t(s,t)\|$. The middle column displays the local maxima of the gradient magnitude (left column). The right column displays the actual segment boundaries as vertical lines (section headings for Wikipedia and author change in Google Wave). The gradient maxima corresponding to vertical lines in the middle column matches nicely the Wikipedia section boundaries. The gradient maxima corresponding to horizontal lines in the middle column correspond nicely to major revisions indicated by a discontinuities in the location of the section boundaries.

we may view segment boundaries as edges in the space-time domain Ω. These boundaries separate the segments from each other, much like borders separate countries in a two dimensional geographical map.

Assuming all edges are correctly identified, we can easily identify the segments as the interior points of the closed boundaries. In general, however, attempts to identify segment boundaries or edges will only be partially successful. As a result predicted edges in practice are not closed and do not lead to interior segments. We consider now the task of predicting segment boundaries or edges in Ω and postpone the task of predicting a segmentation to the next section.

Edges, or transitions between segments, correspond to abrupt changes in the local word distribution. We thus characterize them as points in Ω having high gradient value. In particular, we distinguish between vertical edges (transitions across document positions), horizontal edges (transitions across versions), and diagonal edges (transitions across both document position and version). These three types of edges may be diagnosed based on the magnitudes of $\dot{\gamma}_s$, $\dot{\gamma}_t$, and $\dot{\alpha}_1\gamma_s + \dot{\alpha}_2\gamma_t$ respectively.

Besides the synthetic data results in Figure 12, I conducted edge detection experiments on six different real world datasets. Five datasets are Wikipedia.com articles: Atlanta, Religion, Language, European Union, and Beijing. Religion and European Union are version-controlled documents with relatively frequent updates, while Atlanta, language, and Beijing have less frequent changes. The sixth dataset is the Google Wave Amazon Kindle FAQ which is a less structured version-controlled document.

Preprocessing included removing html tags and pictures, word stemming, stopword removal, and removing any non alphabetic characters (numbers and punctuations). The section heading information of Wikipedia and the information of author of each posting in Google Wave is used as ground truth for segment boundaries. This information was separated from the dataset and was used for training and evaluation

Table 6: Test set error rate and F1 measure for edge prediction (section boundaries in Wikipedia articles and author change in Google Wave). The space-time domain Ω was divided to a grid with each cell labeled edge ($y = 1$) or no edge ($y = 0$) depending on whether it contained any edges. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to the TextTiling test segmentation algorithm [21] without paragraph boundaries information. Method c corresponds to a logistic regression classifier whose feature set is composed of statistical summaries (mean, median, max, min) of $\dot{\gamma}_s(s,t)$ within the grid cell in question as well as neighboring cells.

Article	**Rev.**	**Voc. Size**	$p(y)$	**Error Rate**			**F1 Measure**		
				a	b	c	a	b	c
Atlanta	2000	3078	0.401	0.401	0.424	0.339	0.000	0.467	0.504
Religion	2000	2880	0.403	0.404	0.432	0.357	0.000	0.470	0.552
Language	2000	3727	0.292	0.292	0.450	0.298	0.000	0.379	0.091
European Union	2000	2382	0.534	0.467	0.544	0.435	0.696	0.397	0.663
Beijing	2000	3857	0.543	0.456	0.474	0.391	0.704	0.512	0.682
Amazon Kindle FAQ	100	573	0.339	0.338	0.522	0.313	0.000	0.436	0.558

(on testing set).

Figure 13 displays a gradient information, local maxima, and ground truth segment boundaries for the version controlled Wikipedia articles Religion and Atlanta. The local gradient maxima nicely match the segment boundaries which lead us to consider training a logistic regression classifier on a feature set composed of gradient value statistics (min, max, mean, median of $\|\dot{\gamma}_s(s,t)\|$ in the appropriate location as well as its neighbors (the space-time domain Ω was divided into a finite grid where each cell either contained an edge ($y = 1$) or did not ($y = 0$)). Table 6 displays the test set accuracy and F1 measure of three predictors: our logistic regression (method c) as well as two baselines: predicting edge/no-edge based on the marginal $p(y)$ distribution (method a) and TextTiling (method b) [21] which is a popular text segmentation algorithm. Since I do not assume paragraph information in the experiment, I ignored this component and considered the document as a sequence with $w = 20$ and 29

minimum depth gaps parameters (see [21]). I conclude from the figure that the gradient information leads to better prediction than TextTiling (on both accuracy and F1 measure).

4.2.5 Segmentation

As mentioned in the previous section, predicting edges may not result in closed boundaries. It is possible to analyze the location and direction of the predicted edges and aggregate them into a sequence of closed boundaries surrounding the segments. I take a different approach and partition points in Ω to k distinct values or segments based on local word content and space-time proximity.

For two points $(s_1, t_2), (s_2, t_2) \in \Omega$ to be in the same segment, I expect $\gamma(s_1, t_1)$ to be similar to $\gamma(s_2, t_2)$ and for (s_1, t_1) to be close to (s_2, t_2). The first condition asserts that the two locations discuss the same topic. The second condition asserts that the two locations are not too far from each other in the space time domain. More specifically, I propose to segment Ω by clustering its points based on the following geometry

$$d((s_1, t_1), (s_2, t_2)) \quad = \quad d_H(\gamma(s_1, t_1), \gamma(s_2, t_2)) \; + \; \sqrt{c_1(s_1 - s_2)^2 + c_2(t_1 - t_2)^2} \quad (25)$$

where $d_H : \mathbb{P}_V \times \mathbb{P}_V \to \mathbb{R}$ is Hellinger distance

$$d_H^2(u, v) = \sum_{i=1}^{V} (\sqrt{u_i} - \sqrt{v_i})^2. \quad (26)$$

The weights c_1, c_2 are used to balance the contributions of word content similarity with the similarity in time and space.

Figure 14 displays the ground truth segment boundaries and the segmentation results obtained by applying k-means clustering ($k = 11$) to the metric (25). The figure shows that the predicted segments largely match actual edges in the documents even though no edge or gradient information was used in the segmentation process.

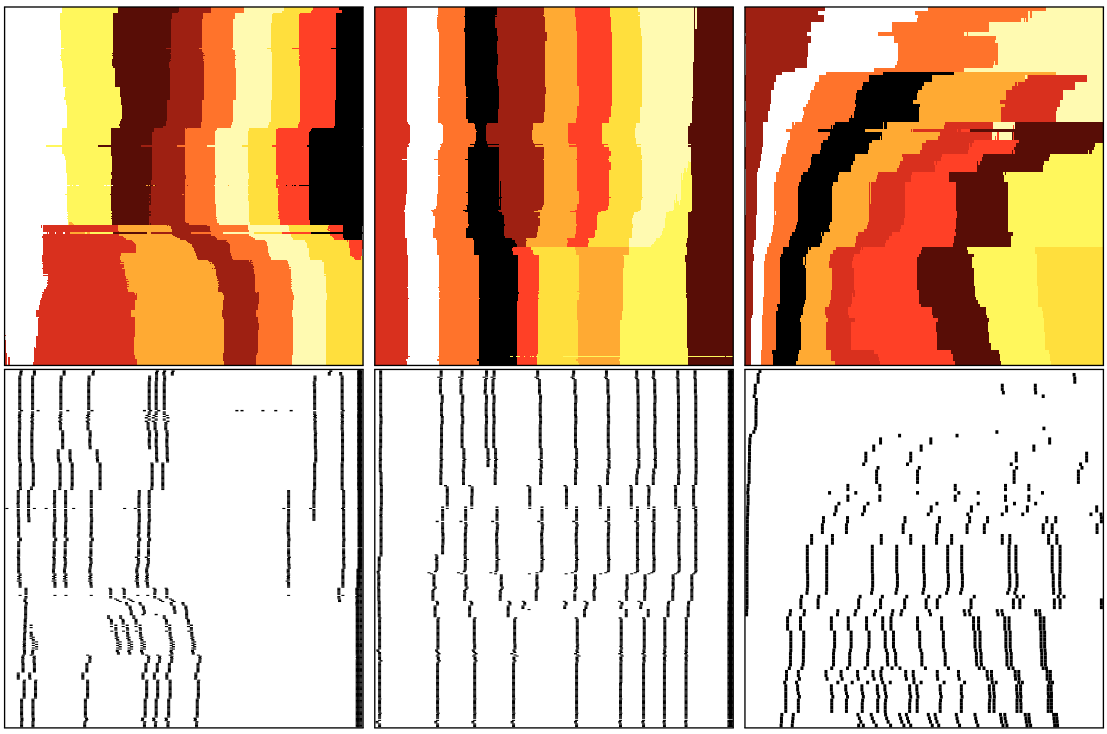

Figure 14: Predicted segmentation (top) and ground truth segment boundaries (bottom) of portions of the version controlled Wikipedia articles Religion (left), Atlanta (middle) and the Google Wave Amazon Kindle FAQ(right). The predicted segments match the ground truth segment boundaries. Note that the first 100 revisions are used in Google Wave result. The proportion of the segments that appeared in the beginning is keep decreasing while the revisions increases and new segments appears.

Table 7: Error rate and F1 measure over held out test set of predicting future UNDO operation in Wikipedia articles. Method a corresponds to a predictor that always selects the majority class. Method b corresponds to a logistic regression based on the term frequency vector of the current version. Method c corresponds a logistic regression that uses summaries (mean, median, max, min) of $\|\dot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_s(s,t)\|$, $g(t)$, and $h(s)$.

Article	**Rev.**	**Voc. Size**	$p(y)$	**Error Rate**			**F1 Measure**		
				a	b	c	a	b	c
Atlanta	2000	3078	0.218	0.219	0.313	0.212	0.000	0.320	0.477
Religion	2000	2880	0.123	0.122	0.223	0.125	0.000	0.294	0.281
Language	2000	3727	0.189	0.189	0.259	0.187	0.000	0.334	0.455
European Union	2000	2382	0.213	0.208	0.331	0.209	0.000	0.275	0.410
Beijing	2000	3857	0.137	0.137	0.219	0.136	0.000	0.247	0.284

4.2.6 Predicting Future Operations

The fourth and final task is predicting a future revision d_{l+1} based on the smoothed representation of the present and past versions $d_1, \ldots, d_l$. In terms of Ω, this means predicting features associated with $\gamma(s,t), t \geq t'$ based on $\gamma(s,t), t < t'$.

I concentrate on predicting whether Wikipedia edits are reversed in the next revision. This action, marked by a label UNDO or REVERT in the Wikipedia API, is important for preventing content abuse or removing immature content (by predicting ahead of time suspicious revisions).

We predict whether a version will undergo UNDO in the next version using a support vector machine based on statistical summaries (mean, median, min, max) of the following feature set $\|\dot{\gamma}_s(s,t)\|$, $\|\ddot{\gamma}_s(s,t)\|$, $\|\dot{\gamma}_t(s,t)\|$), $\|\dot{\gamma}_t(s,t)\|$, $g(h)$, and $h(s)$.

Table 7 shows the test set error and F1 measure for the logistic regression based on the smoothed space-time representation (method c), as well as two baselines. The first baseline (method a) predicts the majority class and the second baseline (method b) is a logistic regression based on the term frequency content of the current test version. Using the derivatives of γ, we obtain a prediction that is better than

choosing majority class or logistic regression based on word content. I thus conclude that the derivatives above provide more useful information (resulting in lower error and higher F1) for predicting future operations than word content features.

4.2.7 Summary and Discussion

The task of analyzing and visualizing a version-controlled document is important because it allows large scale monitoring of collaboratively authored resources such as Wikipedia, GIT and SVN. This framework is the first to develop analysis and visualization tools for this setting. It presents a new representation for version-controlled documents that uses local smoothing to map a space-time domain Ω to the simplex of tf-vectors $\mathbb{P}_V$. I demonstrated the applicability of the representation for four tasks: visualizing change, predicting edges, segmentation, and predicting future revision operations.

Visualizing changes may highlight significant structural changes for the benefit of users and help the collaborative authoring process. Improved edge prediction and text segmentation may assist in discovering structural or semantic changes and their evolution with the authoring process. Predicting a future operation may assist authors as well as prevent abuse in coauthoring projects such as Wikipedia.

The experiments described in this section were conducted on synthetic, Wikipedia and Google Wave articles. They show that the proposed formalism achieves good performance both qualitatively and quantitatively as compared to standard baseline algorithms.

It is intriguing to consider the similarity between the proposed representation and image processing. Predicting segment boundaries is similar to edge detection in images. Segmenting version-controlled documents may be reduced to image segmentation. Predicting future operations is similar to completing image parts based on the remaining pixels and a statistical model. Due to its long and successful history,

image processing is a good candidate for providing useful tools for version-controlled document analysis. The presented framework facilitates this analogy and I believe is likely to result in novel models and analysis tools inspired by current image-processing paradigms. A few potential examples are wavelet filtering, image compression, and statistical models such as Markov random fields.

With respect to the *good representation* criteria in Section 1.4, the local space-time smoothing representation has improved various aspects.

1. Reconstruction Quality: Unlike traditional document representations, the local-space time smoothing representation additionally preserves two types of sequentiality: spatial and temporal.

2. Discriminative Power: With help from richer sequential features, the representation showed stronger discriminative power on detecting edges, segmentation, and predicting an abuse (Table 6,7).

3. Interpretability: Based on gradient fields of the representation, I visualized the degree of change of a document both spatially and temporally (Figure 11,12,13).

4. Computation: The representation is based on two-dimensional kernel smoothing, which can be computed efficiently by Fourier-transform-based convolution methods (by Convolution Theorem).

4.3 Local Sequential Information

4.3.1 Locality of Documents

Learning a representation that reflects word locality is important in a wide variety of text processing applications such as text categorization, information retrieval, or language model generation. The n-gram model, for example, is popular because of its simplicity and efficiency, which interprets a document as a collection of word sub-sequences. Specifically, it models a word given the previous $n-1$

words: $p(w_i|w_{i-1}, \ldots, w_{i-n+1})$. The larger n is, the longer the contexts that the model can capture. A related approach is to model a symmetric window around a word $p(w_i|w_{i+1}, w_{i-1}, w_{i+2}, w_{i-2}, \ldots)$, as is done for example by [52].

[39] extended local dependencies by applying different weights at each position of a document and summing up the word presence near a particular location. Specifically, that approach, named "locally weighted bag-of-words" (LOWBOW), uses a smoothing kernel to generate a smooth curve in the probability simplex that represents the temporal progression of the document. LOWBOW allows examining much longer-range dependencies than n-gram models, and it also allows tying word patterns to specific document locations. The bandwidth of the smoothing kernel captures the tradeoff between estimation bias and estimation variance. The approach in this section extends their work, but is different as it decouples local probabilities from their positions and it uses sparse coding to compress the parameter space.

Document models such as the n-gram and LOWBOW suffer from intrinsic sparsity, an inevitable consequence of capturing dependencies in sequences over a large vocabulary. The larger the dependency range, the harder it is to estimate the dependencies due to increased estimation variance. Specifically, the number of possible combinations of n consecutive words grows exponentially, making the number of observations for each combination extremely sparse, eventually causing not only computational difficulties but also a high estimation error. As a result, in many cases where data is limited, n-gram models with low n perform better than n-gram models with high values of n.

Neural probabilistic language models such as [3] are an attempt to handle this issue. They capture long term relations over a large vocabulary by using a parametric model that compresses the parameter space. Since the model estimates a compressed parameter vector rather than the exponentially growing n-gram counts, it is an effective way of capturing word dependencies that n-gram models cannot. In deep

learning communities, Recurrent Neural Network (RNN) models attempt to capture sequential information in their recurrent neural network connections. For example, LSTM [22] has memorizing capabilities to model long or short term sequential dependencies although it needs a significant amount of computation (see Section 2.2.2 for further discussion). On the other hand, probabilistic topic models such as [7] and matrix decomposition models [14, 40, 88] estimate a compressed representation of the vocabulary, usually termed latent space or topics. Unlike the neural language model, these models are usually based on the bag-of-words representation or bigram features [78], limiting their potential to capture sequential word dependencies (though some recent extensions generalize topic models to sequential models - see Section 2.1.5).

By efficiently estimating sparse and compact representations of local dependencies, my model extends the work of [39] and [88]. I first define the notion of a *local context*, which is a conditional word probability given the word's location in the document. Similar to [39], I use a smoothing kernel to estimate the local context. Each kernel bandwidth examines a unique range of local resolutions. As noted earlier, because of the huge number of local contexts in this model, I apply a sparse-coding formulation to compress the space.

My model has several benefits. First, by introducing rich local dependencies, it can generate highly discriminating features. Second, it produces a sparse and compact representation of a document. Third, since it also models word proximities, it can be used to generate locally coherent topics that will be a useful tool for analyzing the topical flow of a document.

4.3.2 Local Context

Most document and topic modeling studies use sequential features such as unigram or n-gram to model documents. Instead, [39] modeled a document as a joint distribution of words and their locations $p(w, t)$, where w is a word and t is the location. The joint

distribution $p(w,t)$, estimated by kernel density estimation, models the probability that a word will occur at a specific index within the document. Although the approach is useful for modeling document progression, it cannot model the relative positioning of words. On the other hand, $p(w|t)$ can model the relative positioning of words.

A *local context* is the distribution of words that occurred near a specific document position: $p(w|t)$. I denote it by $\phi(t)$:

$$\phi : \mathbb{N} \to \mathbb{R}^{|V|} \quad \text{where } |V| \text{ is the size of vocabulary.}$$

Given a length L document $x = [w_1, \ldots, w_L]$ and a position i, we can estimate the local context $\phi(i)$ using a smoothing kernel $k(i,j)$ that is a real valued normalized function that is monotonic decreasing in $|i-j|$. Intuitively, the kernel defines the locality that we are interested in.

$$\begin{aligned}
\phi(i) &= \left[\phi_1(i), \ldots, \phi_{|V|}(i)\right]^\top \\
\phi_v(i) &= \sum_{j=1}^{L} k(i,j)\mathbf{1}_{\{w_j=v\}}. \\
&\text{s.t. } \sum_v \phi_v(i) = 1, \;\; \forall_v \; \phi_v(i) \geq 0
\end{aligned}$$

4.3.2.1 Choices of $k(i,j)$

There are several standard choices of a smoothing kernel $k(i,j) = g(i-j)$. I follow [39] and use the Gaussian kernel, which is a normalized Gaussian density. However, for illustration purposes, I use below the constant kernel (for a support of 3 words)

$$k'(i,j) = \begin{cases} 1/3 & \text{if } |i-j| \leq 1 \\ 0 & \text{else.} \end{cases}$$

This kernel measures the existence of a word in the window $\{w_{i-1}, w_i, w_{i+1}\}$. It differs from the trigram representation in that it ignores the ordering within the window.

Non-constant kernels such as Gaussian kernels allow emphasizing words closer to the center of the window while discounting more remote locations.

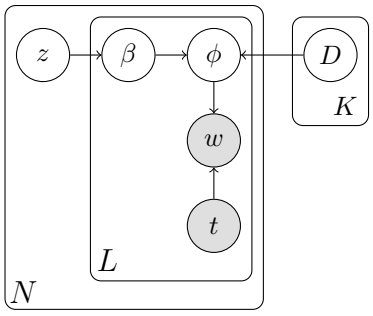

Figure 15: Graphical model of local context sparse coding. z denotes a document representation, and ϕ denotes a local context in a document of length L. D is a shared dictionary (topics), and β is a latent representation of a corresponding local context using D. See Section 4.3.3.1 for details.

4.3.2.2 Comparision with n-gram Models

The n-gram model and its variations fundamentally differ from this model since they use a joint distribution of consecutive words, $p(w_i, \ldots, w_{i-n+1})$, instead of a conditional distribution between words and locations, $p(w|t)$. The size of the event space of an n-gram model expands exponentially when either its vocabulary or the size of window (n) grows. By contrast, the event space of this model is invariant of the window size (or the kernel bandwidth) and only linear in vocabulary size. In practice, the n-gram model performs poorly when both the vocabulary and n are large. See Section 4.3.4.3 for empirical results.

4.3.3 Local Context Sparse Coding (LCSC)

We now consider the bag of local contexts of a document, $\Phi = \{\phi(i) : i = 1, \ldots, L\}$ (where L is the length of the document). Since direct estimation of bag-of-local-contexts statistics is intractable, I approximate each $\phi(i)$ using a linear combination of a handful (sparse) of codes in a dictionary of K codes (or *topics*).

$$\phi(i) \approx D\beta(i) \text{ where } D \in \mathbb{R}^{|V| \times K}, \beta(i) \in \mathbb{R}^K$$
$$\text{s.t. } \beta(i) \text{ is sparse, } D \geq 0, \forall_i \sum_j D_{ij} = 1$$

Note in particular that the dictionary can be shared across multiple documents, and as a result the β that corresponds to different Φ (documents) are comparable.

I measure the approximation quality using the sum of squared distances between each $\phi(i)$ and $D\beta(i)$ and add a L_1 penalty on $\beta(i)$ to enforce sparsity. This standard practice is equivalent to maximizing the penalized likelihood of the model under a Gaussian distribution (regression) with a Laplace prior $p(\beta) \propto e^{-\lambda|\beta|}$ corresponding to the L_1 penalty. Thus, we get the following objective function for learning the dictionary D and the β parameters

$$\sum_{i=1}^{L} \|\phi(i) - D\beta(i)\|_2^2 + \lambda\|\beta(i)\|_1 \tag{27}$$

subject to the constraints of $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. When we have multiple documents, we combine multiple squared error terms where the D matrix is shared and β parameters correspond to different documents as in (31).

Alternatively, we can use non-squared error loss functions as in [41]. In the following experiments, we used the Hellinger distance $\|\sqrt{\phi(i)} - \sqrt{D\beta(i)}\|_2^2$, which performed the best. See [35, 36, 15] for additional examples of using Hellinger distances in text modeling and interpreting it in terms of information geometry.

I assume that the topic assignment parameters for a specific document are normally distributed $\beta(i)|z \sim \mathcal{N}(z, \rho^{-1}\mathsf{I})$ and consider its mean z as a document-specific parameter, or a *document representation*. This leads to the above objective function

$$\sum_{i=1}^{L} \rho\|\beta(i) - z\|_2^2 + \|\phi(i) - D\beta(i)\|_2^2 + \lambda\|\beta(i)\|_1 \tag{28}$$

subject to $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. The equations above assume a single document. In the case of multiple documents, we sum over them as described in Section 4.3.3.2. In this case, D is shared across documents and β and z are document specific.

4.3.3.1 Comparison with Probabilistic Topic Models

The proposed method forms a graphical model as described in Figure 15, with the details appearing below. I follow some of the ideas in [88] and note the caveat that

the normalization in my model may not be consistent with the true distribution generating the data due to the fact that the parameters lie in a restricted domain (see comment below).

1. The local probability of words (or a local context) follows a distribution centered on $D\beta$ where D contains topics shared across multiple documents and β contains a corresponding topic assignment. For example, assuming a Gaussian distribution, we have:

$$\phi = p(w|t) \sim \mathcal{N}(D\beta, \sigma_\phi \mathrm{I}). \tag{29}$$

2. Topic assignments parameters $\{\beta(i) : i = 1, \ldots, L\}$ that correspond to a specific document follow a normal distribution centered on z with a Laplace prior.

$$\beta|z \sim \mathcal{N}(z, \rho^{-1}\mathrm{I}), \quad \beta \sim \text{Laplace}(0, \lambda^{-1}) \tag{30}$$

Traditional probabilistic topic models differ from my model primarily in two ways. First, instead of a single word observation $p(w)$, I model word locality through the distribution $p(w|t)$. Second, I do not directly compute the normalization terms of each probabilistic distribution. I only compute the numerator, for example $\|\beta(i) - z\|_2^2$, which is consistent with a Gaussian distribution but ignores the fact that β cannot achieve all values in a Euclidean space. This relaxation reduces the overall computation when compared to standard probabilistic topic models.

4.3.3.2 Estimation

The training procedure of LCSC model is similar to the one of standard sparse coding models. Assuming we have multiple documents $X = [x^{(1)}, \ldots, x^{(N)}]$, we minimize the aggregated loss function of (28),

$$\min_{\beta,z,D} \ell = \min_{\beta,z,D} \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \left[\rho \|\beta^{(n)}(i) - z^{(n)}\|_2^2 + \|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2 + \lambda \|\beta(i)\|_1 \right] \tag{31}$$

subject to the following constraints on the shared dictionary D: $D \geq 0$ and $\forall_i \sum_j D_{ij} = 1$. It is a biconvex problem that can be iteratively solved for β, z and D. We additionally include non-negativity constraint on β for better interpretability, similar to [88].

Solving for β and z By repeatedly optimizing each dimensions of β (coordinate descent), the lasso problem can be solved in closed form and have a unique solution under the non-negativity constraint. Specifically, using the shorthand notation $\beta^{(n)}(i) \to \beta$, $z^{(n)} \to z$, $\phi^{(n)}(i) \to \phi$, minimizing a single component of $\beta^{(n)}(i)$ gives the following:

$$\begin{aligned}
&\min_{\beta_j} \sum_{k=1}^{K} \rho(\beta_k - z_k)^2 + \sum_{v=1}^{|V|} \left(\phi_v - \sum_{k=1}^{K} D_{vk}\beta_k \right)^2 + \sum_{k=1}^{K} \lambda|\beta_k| \\
&= \min_{\beta_j} \left[\underbrace{\left(\rho + \|D_{:j}\|_2^2\right)}_{a} \beta_j^2 - 2 \underbrace{\left(\rho z_j + \sum_{v=1}^{|V|} D_{vj} \left(\phi_v - \sum_{k \neq j} D_{vk}\beta_k \right) \right)}_{b} \beta_j + \lambda|\beta_j| \right].
\end{aligned}$$

The corresponding optimal solution is

$$\beta_j = \frac{1}{a} \min\left(0, b - \frac{\lambda}{2}\right). \tag{32}$$

The corresponding document representation $z^{(n)}$ also can be solved in closed form since we are minimizing L_2 distances between $z^{(n)}$ and $\beta^{(n)}(1), \ldots, \beta^{(n)}(L^{(n)})$.

$$z^{(n)} = \frac{1}{L^{(n)}} \sum_i \beta^{(n)}(i). \tag{33}$$

We would normally iterate the dimensions of β in a sequential order ($j = 1, 2, \ldots, K$) until convergence, which is called pathwise coordinate descent as was done in the training of STC [88]. Greedy coordinate descent [45], however, updates one dimension at a time by choosing the dimension that reduces the loss the most ($\Delta\ell$). This results in faster training than pathwise method with the same accuracy level. See [45] for detailed discussion.

Algorithm 1 Greedy coordinate descent for β and z

Input: local contexts of $x^{(1)}, \ldots, x^{(N)}$ and D
for all $x \in \{x^{(1)}, \ldots, x^{(N)}\}$ **do in parallel**
  $\Phi = [\phi(1), \ldots, \phi(L)]$ in x
  $[b(1), \ldots, b(L)] = D^\top \Phi$
  $z = \frac{1}{L} \sum_i b(i)$

  while $\sum_i |\beta(i)^{t+1} - \beta(i)^t| > \epsilon$ **do**
    $z^{t+1} = z^t$
    for all $i \in \{1, \ldots, L^{(n)}\}$ **do in parallel**
      $\tilde{\beta}(i) = \frac{1}{a} \min(0, b(i) - \lambda/2)$
      $j = \arg\max_k |\tilde{\beta}(i)_k - \beta(i)^t_k|$

      $$\beta(i)^{t+1} = \begin{cases} \tilde{\beta}(i)_j & \text{at } j\text{th dimension} \\ \beta(i)^t & \text{else} \end{cases}$$

      $z_j^{t+1} = z_j^t + (\beta_j^{t+1} - \beta_j^t)/L$
      $b(i)^{t+1} = b(i)^t - (\beta(i)_j^{t+1} - \beta(i)_j^t)(D^\top D) e_j$

      # wait for others to finish updating z
      $b(i)_j^{t+1} = b(i)_j^t + \rho(z_j^{t+1} - z_j^t)$
    end for
  end while
end for
Output: $z^{(1)}, \ldots, z^{(N)}$ and all β for all local contexts.

By applying greedy coordinate descent and exploiting the factorization of the loss function, I developed an efficient algorithm for β and z (see Algorithm 1). Since greedy coordinate descent ensures the difference between β^{t+1} and β^t is exactly $\beta_j^{t+1} - \beta_j^t$ (j is the updated dimension), b and z can be updated efficiently using the previous values of those. In addition, β and z of a document are independent from those of other documents, and $\{\beta(i) : i = 1, \ldots, L\}$ in a single document only shares z during the update, which allows parallelization. Note that we approximate the loss decrease $\Delta\ell$ by $|\tilde{\beta}(i) - \beta^t(i)|$ (see Algorithm 1 for details.)

Solving for D Projected gradient descent method efficiently optimizes the dictionary D under the simplex constraint ($D \geq 0,\ \forall_i \sum_j D_{ij} = 1$).

$$\min_D \ell(D) = \min_D \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2 \tag{34}$$

$$\nabla \ell(D) = -2 \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \Big[\phi^{(n)}(i) - D\beta^{(n)}(i)\Big] \beta^{(n)}(i)^\top \tag{35}$$

Specifically, we take a gradient step based on the gradient above and then project back to the simplex using a simplex projection Π.

$$D^{t+1} = \Pi(D^t - \eta_t \nabla). \tag{36}$$

The projection Π can be computed efficiently, see for example [16] for details. We estimate the step size η by a line search that minimizes the dictionary related loss $\min_\eta \sum \|\phi - D^{t+1}\beta\|_2^2$.

4.3.4 Experiments

4.3.4.1 Illustrating Example

I illustrate the proposed method (LCSC) using a synthetic example of four documents with two different types of word locality: $\{a, b\}$ vs $\{a, c\}$.

$$x_1 = [a, b, a, b, a, b, c, c, c], \quad x_2 = [b, a, b, a, b, a, c, c, c]$$
$$x_3 = [a, c, a, c, a, c, b, b, b], \quad x_4 = [c, a, c, a, c, a, b, b, b]$$

While a and b accompany together in x_1 and x_2, a and c are together in x_3 and x_4, resulting in the topics of x_1 and x_2 being different from the topics of x_3 and x_4.

Bag-of-words representation, a common feature for topic models, generates exactly the same representations $[3, 3, 3]$ or $[0.33, 0.33, 0.33]$ (normalized) for all documents. By contrast, the bigram model distinguishes all four documents although it strictly separates two locally similar pairs ($[a, b]$ and $[b, a]$) at the same time. Despite the fact that the strict separation might be a preferable choice, this will eventually lead to

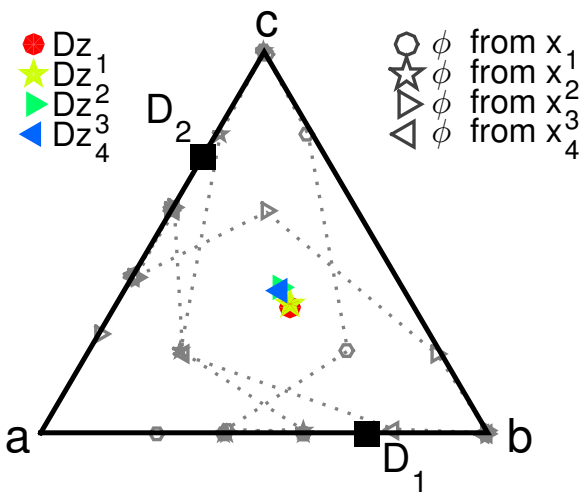

Figure 16: Result of LCSC on the synthetic example of Section 4.3.4.1 in a simplex, each corner of which represents the probability of one of the corresponding character. Filled shapes (Dz) denote document representations on the simplex; unfilled shapes (ϕ) are for local contexts of each document; filled squares are for two topics D_1, D_2. We see clear separation between $\{Dz_1, Dz_2\}$ vs $\{Dz_3, Dz_4\}$.

an explosion of the feature space (especially when trying to account for long-range dependencies). See Section 4.3.2.2 for detailed discussion.

Unlike n-gram models, LCSC easily captures two topics corresponding to two distinct types of locality. Figure 16 shows the result of LCSC in a simplex using a dictionary of size $K = 2$ (number of topics) and a Gaussian smoothing kernel with bandwidth of 0.7. The smoothing kernel covers an effective width of about 5 words (weighted non-uniformly).

Figure 16 visualizes the characteristics of the dataset. First, two topics D_1 and D_2 capture two different types of locality. D_1 is located between a and b denoting the mixture topic of a and b; D_2 is located between a and c. Second, document representations on the simplex (Dz) form two separate groups. The first group consists of Dz_1 and Dz_2 and the second group consists of Dz_3 and Dz_4. The positions of the document representations discriminate documents by its local word distribution $p(w|t)$. Note that n-gram model cannot easily achieve this.

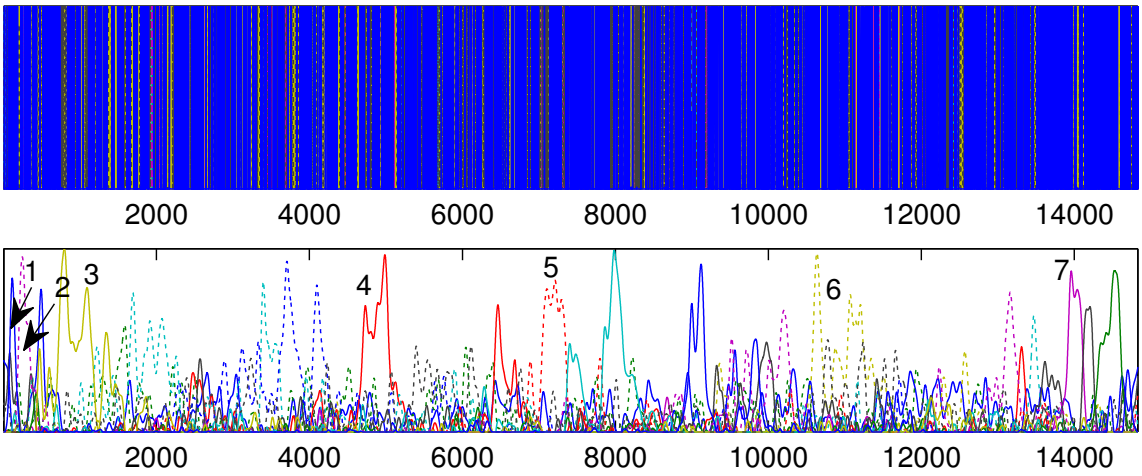

Figure 17: Topic assignments at each position of Wikipedia article "Paris" by LDA (top) and LCSC (bottom). The leftmost edge indicates the beginning of the document and the rightmost edge for the end. Each line type indicates a single topic with its vertical position as a corresponding topic strength. LCSC topics are more locally distributed than LDA. Numbers on the bottom figure indicate topic IDs; Table 8 has the detail of each topic.

4.3.4.2 Local Topics

In contrast to the topics of traditional topic models, LCSC topics reflect the word locality. For instance, Latent Dirichlet Allocation (LDA) [7] will fail to capture any meaningful topics on the synthetic example of Section 4.3.4.1 because all four documents have the same uniform word distribution. Unlike LDA, LCSC discovered two topics corresponding to two distinct types of locality in the previous section. In addition, as each local context contains its neighborhood information, LCSC eventually forms locally coherent topics, which are useful in practice since most text in general have locally coherent contents.

I compare LCSC with a well known topic-modeling technique, LDA, on a real world data: a Wikipedia article "Paris." I chose the article because it contains common knowledge and is well structured, albeit I do not use any structural information.

Figure 17 shows topic assignments at each position of the Paris article by LDA and LCSC (K=15 for both). The document progresses from left to right and each

Table 8: Top words of selected topics using LCSC on a Wikipedia article "Paris." See text for details.

1	mi km sq area population kilometres bois city north paris river climate arrondissements vincennes south
2	world fashion paris international high cent largest manufacturing business million europe region global
3	roman bc parisii century found seine bank romans lutetia ad left le site cit soldiers age excavations built
4	king national government july commune paris sans culottes city army guard palace festival revolution
5	exposition champs universal visitors eiffel tower mars meters held world palais million iii hosted place
6	theatre arrondissement des tel du located musee district ra including centre op paris place theatres lies
7	library paris arrondissement libraries le biblioth university public located sorbonne mitterrand ois fran

position corresponds to a word. The top figure (LDA) does not show any locally coherent structure, which is rather fragmented into pieces. In the bottom figure (LCSC), the topic assignments are locally coherent and illustrate the semantic flow of the document; it starts with the introduction of the city: general information (topic 1 on Table 8) and its reputation (topic 2), which are followed by several aspects of Paris: history (topic 3,4), exposition (topic 5), art (topic 6), and education (topic 7). In addition, top words of each topic are indeed highly indicative of each local subject (Table 8).

I also tried other types of documents that are not structurally written, such as novels ("The Metamorphosis" by Kafka, "The Last Leaf" by O. Henry), a speech ("I Have a Dream" by MLK), and an editorial (a Watergate article), and they all demonstrated an ability to learn locally coherent topics.

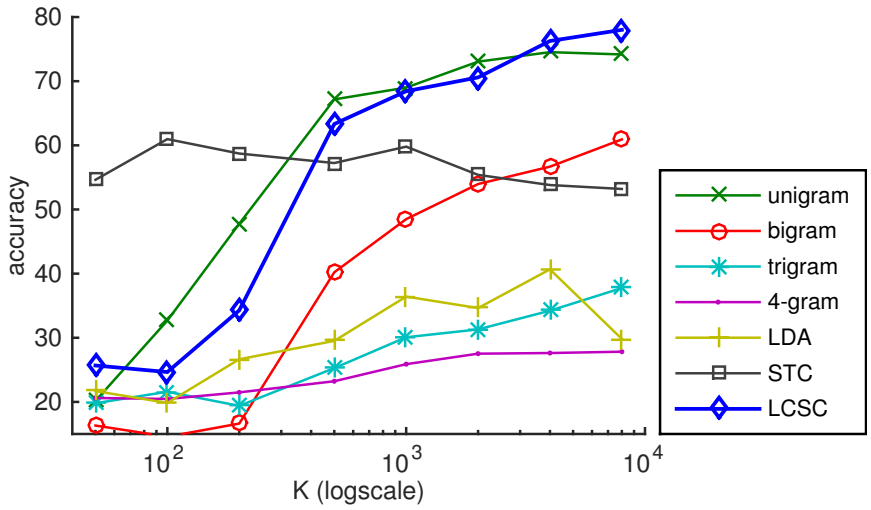

Figure 18: Test set classification accuracies with various dictionary sizes (K) and methods (different line styles)

4.3.4.3 Classification

I examine in this section using features generated by LCSC in classification. I used a standard classifier, support vector machine[2], with different sets of features. Specifically, I used ν-SVM whose ν value was selected from 10 candidate values using cross-validation.

The classification task is based on standard 20 newsgroup[3] classification data with the official train-test split and standard tokenizing sentences and words, Porter stemming, and removing rare features and stop words. The preprocessing resulted in 18846 documents, 20 classes, and vocabulary of size $|V| = 6328$. In the following two subsections, to examine the effect of parameters, I handle a subset of the dataset (5 classes, comp.*). In the last subsection, I evaluate overall performance on both the subset of the dataset and the whole dataset.

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[3] `http://qwone.com/~jason/20Newsgroups/`

Effect of the Number of Topics (K) Figure 18 shows test set classification accuracies with various methods and sizes of dictionaries (from 50 to 8000). In the case of n-gram models, I selected the most frequent K features from the training set. For the other methods LDA[4], STC[5], and LCSC, I specified the size of a dictionary as a parameter. The bandwidth of LCSC was fixed to h=1, which covers about 7 words ($\pm 3h$). I tried a set of candidates for the remaining parameters and chose the best performing one (for example, $\lambda = \{10^{-4}, 10^{-2}, 10^{-1}, 0.5, 1\}$ for STC).

LCSC performs similar to unigram with small dictionaries, but it eventually achieves superior performance with a dictionary of sufficient size (from K=4000), that is, the performance of LCSC keeps improving even after $K>|V|$ (unigram model reaches maximum performance when $K<|V|$). STC performs well with relatively small dictionaries, but its maximum performance is not as good as other methods.

Figure 18 partially confirms Section 4.3.2.2. Bigram, trigram and 4-gram model do not perform well even with a large dictionary. It is because the number of features grows rapidly (bigram generates $23|V|$ features, trigram for $35|V|$, and 4-gram for $37|V|$) and thus will drastically lower the number of observations for each feature. On the contrary, even though LCSC covers approximately 7 neighboring words, it does not seem to suffer from sparsity and shows superior performance.

Effect of Bandwidth (h) Figure 19 shows test set classification accuracies of LCSC with various bandwidths while other parameters are fixed (K=4000, ρ=10^{-4}, λ=10^{-2}). The best performance was obtained at h=1. Using narrower bandwidth (h=0.5) led to faster convergence to poor performance, which is caused by lack of variability of local features. Using broader bandwidth (h=4) slowed down the convergence and ruined the performance, which is attributed to including unnecessary local dependencies for this task. The diverse results of various bandwidths confirms

[4]FastLDA: `http://www-users.cs.umn.edu/~shan/mmnb_code.html`
[5]`http://www.ml-thu.net/~jun/stc.shtml`

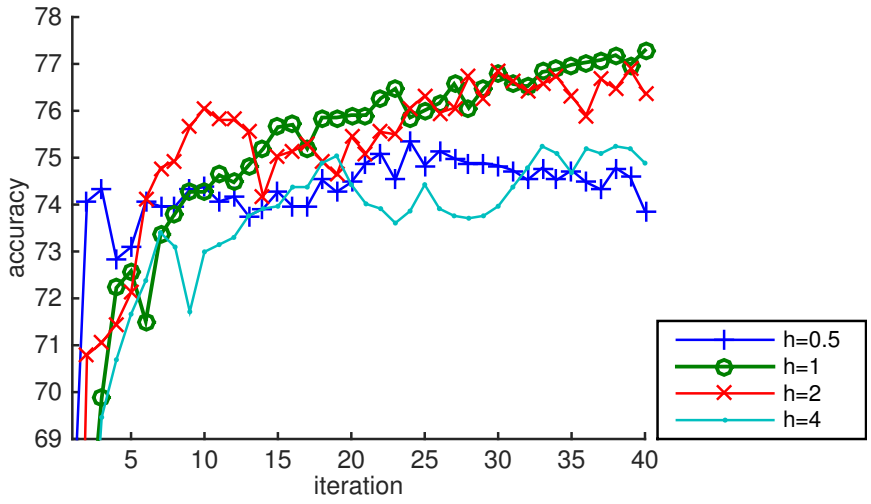

Figure 19: Test set classification accuracies of LCSC with various smoothing bandwidths (h)

Table 9: Comparision of test set classification accuracy for various methods on 5 classes (comp.*) and full 20 classes (*) of 20 newsgroup dataset

	n-gram	LDA	STC	LCSC	MedSTC
comp.*	74.53	40.67	60.97	**78.01**	77.70
*****	74.10	34.43	61.14	**80.76**	79.81

that locality features makes a notable difference in classification performance.

Comparision of Overall Performance I finally compare the overall performance of LCSC with other methods including a local-dependency model, n-gram, and unsupervised topic models: LDA and STC. I additionally included a top performing *supervised* topic model, MedSTC [88]. Note, however, that MedSTC uses auxiliary supervised information (labeled data) during its topic learning, and cannot be directly compared to LCSC. I tried various sets of parameters and choose the best performing one ($K : [50, \ldots, 8000]$, $\lambda, \rho : [10^{-4}, \ldots, 10^{-1}]$). For n-gram models, I tried $n : [1, ..., 4]$ and chose the best.

LCSC outperforms all other competitors on the subset as well as the full set

(Table 9). The performance gain with respect to n-gram models shows that modeling long-range dependencies can be beneficial in classification. The better performance of LCSC compared to other methods including MedSTC (significant at p-value: 0.002) is notable since MedSTC directly optimizes for its discriminative performance whereas LCSC is a purely unsupervised coding method.

4.3.5 Summary and Discussion

This section presented a non-probabilistic topic model for local word distributions. The model employed a kernel smoothing to capture sequential information, which granted a flexible and efficient way to handle a wide range of local information. The sparse-coding formulation leads to efficient training procedures and a sparse representation that is locally coherent and stronger in discrimination.

LCSC is particularly interesting as we have direct control of most of the four *good representation* criteria (Section 1.4). The main loss function (31) contains explicit terms for the criteria (copied below).

$$\min_{\beta,z,D} \ell = \min_{\beta,z,D} \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \left[\underbrace{\rho \|\beta^{(n)}(i) - z^{(n)}\|_2^2}_{(a)} + \underbrace{\|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2}_{(b)} + \underbrace{\lambda \|\beta(i)\|_1}_{(c)} \right]$$

In the formulation, each term evaluates one of the *good representation* criteria. The reconstruction quality of local observations (local contexts) is measured by (b). The discrimination power of resulting representation is controlled by (a) and the interpretability is encouraged or discouraged by (c).

The balance of the three components is adjusted by weighting coefficients: ρ and λ. A large ρ penalizes more on (a) than other losses, which shrinks each β in a document to its center. In its extreme where $\rho = \infty$, all β are constrained to be the same, ignoring the diversity of local features. This lowers the resulting representation's discrimination power. On the other hand, a small ρ increases the discrimination power, but also exposes an overfitting problem because the learning

process is too sensitive on reconstruction quality and sparsity. Besides, a large λ enforces sparser outputs that will be easier to be interpret. Hence, the λ directly controls interpretability of our representation.

Overall, the LCSC representation can be evaluated as the following:

1. Reconstruction Quality: Unlike comparable topic models, the LCSC representation preserves word-level sequentiality, the quality of which is directly controllable by loss (a).

2. Discriminative Power: Because the representation is based on rich local features, it has stronger discriminative power (see Section 4.3.4.1 and Section 4.3.4.3 for experiments). Please note I examine stronger discriminative model in Chapter 5 by employing additional supervised information.

3. Interpretability: The sparsity loss (c) results in sparser representations that are easier to interpret. Moreover, we can extract locally coherent topics that are useful for understanding the semantic flow of a document.

4. Computation: Based on the parallel greedy coordinate descent and the projected gradient descent algorithm, LCSC representation can be learned very efficiently.

4.4 Chapter Discussion

In this chapter, we discussed document representations based on a joint or conditional probability of words and their sequential information. The joint probability of words with their spatial and temporal location effectively modeled the two levels of sequentiality. The conditional probability showed stronger discriminative power and better interpretability by capturing word proximities.

As discussed in this chapter, modeling a document with its sequential information as a multi-dimensional distribution brings a flexible and efficient formulation. In its

center, a kernel plays a significant role conveying a locality concept. A wide kernel results in a representation that focuses on global patterns in a document while a narrow kernel focuses more on local attributes.

The kernels in this chapter were defined in a spatial and temporal domain of a document. Those kernels, which capture word-level, paragraph-level and revision-level sequentiality, can be extended to capture a different notion of sequentiality. For example, an extended kernel additionally based on a coreference distance metric will capture coreference dependencies in the resulting representation. Similarly, other grammatical structures can be utilized to capture a different sequentiality.

In the next chapter, I will extend the LCSC model (Section 4.3) to a unified model incorporating both sequential information and labels.

CHAPTER V

UNIFIED VIEW OF UTILIZING BOTH LABELS AND SEQUENTIAL INFORMATION

5.1 Labels and Sequential Information

In Chapters 3 and 4, I examined document representations based on either labels or sequential information. Both factors were independently useful for obtaining a better representation by the *good representation* criteria (Section 1.4).

In this chapter, I present a unified model that utilizes both labels and sequential information by following the formalism of LCSC model (Section 4.3). Similar to LCSC model, the new model has parameters that directly impacts the balance of *good representation* criteria, which is useful for customizing a representation based on our need.

Labels and sequential information are common in text. Being fundamental to text, sequentiality is found in every text. Labels are also common because we can easily incorporate domain knowledge or annotations as labels. Hence, employing both types of information will be beneficial to a large portion of text analysis applications.

5.2 Supervised Local Topic Modeling

5.2.1 Local Context Sparse Coding Model

In Section 4.3, LCSC model assumed that local histograms of a document, $p(w|t)$ (called local contexts ϕ), are generated by linear combinations of two non-negative matrices that correspond to topics ($D \in \mathbb{R}^{|V| \times K}$) and embeddings ($\beta$). Additionally, I introduced a document representation z that is the center of all embeddings of a document. The formulation constructs the main objective function (31) in Section 4.3.3.2

(copied below).

$$\min_{\beta,z,D} \sum_{n=1}^{N} \sum_{i=1}^{L^{(n)}} \left[\rho \|\beta^{(n)}(i) - z^{(n)}\|_2^2 + \|\phi^{(n)}(i) - D\beta^{(n)}(i)\|_2^2 + \lambda \|\beta(i)\|_1 \right]$$
$$\text{such that } D, \beta \geq 0 \text{ and } \sum_j D_{ij} = 1.$$

This can be simplified using matrix forms where $\Phi^{(n)} = [\phi^{(n)}(1), \ldots, \phi^{(n)}(L^{(n)})] \in \mathbb{R}^{|V| \times L^{(n)}}$ and $B^{(n)} = [\beta^{(n)}(1), \ldots, \beta^{(n)}(L^{(n)})] \in \mathbb{R}^{K \times L^{(n)}}$.

$$\min_{B,z,D} \sum_{n=1}^{N} \left[\rho \sum_i \|B^{(n)}(i,:) - z^{(n)}\|_2^2 + \|\Phi^{(n)} - DB^{(n)}\|_F^2 + \lambda \sum_i \|B(i,:)\|_1 \right] \tag{37}$$
$$\text{such that } D, B \geq 0 \text{ and } \sum_j D_{ij} = 1.$$

(37) minimizes the distance between local observations of a document with a low-rank approximation by D and B, and it encourages sparse representation that benefits from sparsity loss $\lambda \|B(i,:))\|_1$. The constraint on D restricts each topic (rows of D) to lie on a simplex.

5.2.2 Unified Formulation

LCSC representation employed rich local word dependencies that resulted in strong prediction performance although the model was not directly connected to a text's supervised information.

I propose a unified model that additionally employs supervised information to maximize discriminative performances by augmenting a multiclass prediction loss $f_\Theta(z, y)$ in the previous objective function (37).

$$\min_{B,z,D} \sum_{n=1}^{N} \Bigg[\eta f_\Theta(z^{(n)}, y^{(n)}) + \rho \sum_i \|B^{(n)}(i,:) - z^{(n)}\|_2^2 + \|\Phi^{(n)} - DB^{(n)}\|_F^2 + \lambda \sum_i \|B(i,:)\|_1 \Bigg] \tag{38}$$
$$\text{such that } D, B \geq 0 \text{ and } \sum_j D_{ij} = 1.$$

η is a parameter balancing the predictive loss and other losses, and $\Theta \in \mathbb{R}^{|Y| \times K}$ is a parameter for a classifier.

Various types of multiclass loss can be employed for $f_\Theta(z, y)$, but hereby I employ a multiclass Hinge loss function similar to [88].

$$f_\Theta(z, y) = \Delta(\hat{y}, y) + \Theta(\hat{y}, :)z - \Theta(y, :)z \tag{39}$$
$$\text{where } \hat{y} = \max_y \Theta(y, :)z$$

5.2.3 Estimation

The original LCSC estimation procedure performed two block coordinate descent for $\{B, z\}$ and D. As we have an additional loss function for z, $\{B, z\}$ are no longer solved by simple update rules as shown in Algorithm 1 (Section 4.3). We instead perform three block coordinate descent on B, z, and D iteratively.

Algorithm 2 Greedy coordinate descent for only β

Input: local contexts of $x^{(1)}, \ldots, x^{(N)}$, D, and z
for all $x \in \{x^{(1)}, \ldots, x^{(N)}\}$ **do in parallel**
  $\Phi = [\phi(1), \ldots, \phi(L)]$ in x
  $[b(1), \ldots, b(L)] = D^\top \Phi + \rho \cdot z$

  while $\sum_i |\beta(i)^{t+1} - \beta(i)^t| > \epsilon$ **do**
    $z^{t+1} = z^t$
    for all $i \in \{1, \ldots, L^{(n)}\}$ **do in parallel**
      $\tilde{\beta}(i) = \frac{1}{a} \min(0, b(i) - \lambda/2)$
      $j = \arg\max_k |\tilde{\beta}(i)_k - \beta(i)_k^t|$

      $$\beta(i)^{t+1} = \begin{cases} \tilde{\beta}(i)_j & \text{at } j\text{th dimension} \\ \beta(i)^t & \text{else} \end{cases}$$

      $b(i)^{t+1} = b(i)^t - (\beta(i)_j^{t+1} - \beta(i)_j^t)(D^\top D)e_j$
      $b(i)_j^{t+1} = b(i)_j^t$
    end for
  end while
end for
Output: all β for all local contexts.

Solving for β Although decoupling z from β estimation can slightly decrease the accuracy of the solution, it actually provides a much faster algorithm because we do not have a synchronization step that waits for other parallel processes to finish updating z (see Algorithm 1).

Algorithm 2 shows a revised β update algorithm that is much simpler than the previous approach. Unlike Algorithm 1, we do not have to wait during z_t synchronization steps. It is important to note that b is now initialized to $D^\top \phi + \rho \cdot z$ instead of $D^\top \phi$ in the previous algorithm because z is not starting at 0 anymore.

We can also reduce the formulation into a standard Non-negative Matrix Factorization (NMF) problem by relaxing the L_1 sparsity constraint to a L_1^2 constraint. This is useful for utilizing off-the-shelve NMF softwares that are based on standard NMF algorithms such as active-set or Block Principle Pivoting [25]. By substituting $\lambda \sum_i \|B(i,:)\|_1$ to $\lambda \sum_i \|B(i,:)\|_1^2$ on (38), the formulation becomes

$$\min_B \sum_{n=1}^{N} \Bigg[\eta f_\Theta(z^{(n)}, y^{(n)}) + \rho \sum_i \|B^{(n)}(i,:) - z^{(n)}\|_2^2 + \|\Phi^{(n)} - DB^{(n)}\|_F^2 + \lambda \sum_i \|B(i,:)\|_1^2 \Bigg]. \quad (40)$$

$$\text{such that } D, B \geq 0 \text{ and } \sum_j D_{ij} = 1.$$

Removing all irrelevant variables for B optimization, we have

$$\min_{B \geq 0} \sum_{n=1}^{N} \left[\rho \sum_i \|B^{(n)}(i,:) - z^{(n)}\|_2^2 + \|\Phi^{(n)} - DB^{(n)}\|_F^2 + \lambda \sum_i \|B(i,:)\|_1^2 \right], \quad (41)$$

which is equivalent to solving the following NMF problem for all documents ($n = [1, \ldots, N]$).

$$\min_{B^{(n)} \geq 0} \left\| \begin{pmatrix} D \\ \sqrt{\rho} \cdot \mathsf{I}_{K \times K} \\ \sqrt{\lambda} \cdot \mathbf{1}_{1 \times K} \end{pmatrix} B^{(n)} - \begin{pmatrix} \Phi^{(n)} \\ z^{(n)} \cdot \mathbf{1}_{1 \times L} \\ \mathbf{0}_{1 \times L} \end{pmatrix} \right\|_F^2 \quad (42)$$

However, although standard NMF algorithms are much faster when the number of topics K is small ($10 \leq K \leq 100$), the proposed greedy coordinate descent (Algorithm 2) is much faster when K is large ($K \geq 1000$). This is because my algorithm has $O(K)$ time complexity [45] while active-set variant algorithms have $O(K^3)$ time complexity [25]. I will discuss this further in the experiment section (Section 5.2.4.2)

Solving for z Unlike the previous approach in Section 4.3.3.2, we additionally need to optimize Hinge loss (39) for z as the following:

$$\min_{z^{(n)}} \sum_{i=1}^{L} \eta f_\Theta(z^{(n)}, y^{(n)}) + \|B(:,i) - z\|_2^2, \tag{43}$$

In the case of a linear prediction function $f_\Theta(z^{(n)}, y) = \Theta(y,:)z$ and $\hat{y} = \arg\max_y f_\Theta(z^{(n)}, y) = \Theta(y,:)z$, we conveniently have a closed form solution (45) by taking a derivative with respect to z.

$$0 = \sum_{i=1}^{L} \eta \left[\Theta(\hat{y},:) - \Theta(y,:)\right] - 2(B(:,i) - z) \tag{44}$$

$$\text{where } \hat{y} = \max_y \Theta(y,:)z$$

$$z = \frac{1}{L}\sum_{i}^{L} B(:,i) - \frac{\eta}{2}\left[\Theta(\hat{y},:) - \Theta(y,:)\right] \tag{45}$$

The prediction model parameter Θ can be obtained by solving a standard multiclass Hinge loss optimization procedure.

Solving for D The same projected gradient descent update (36) can be utilized to solve constrained D optimization (Section 4.3.3.2).

5.2.4 Experiment

The unified formulation is expected to perform stronger in classification tasks compared to LCSC model because of the additional predictive loss that encourages discriminative representation. Predictive loss coefficient η adjusts how much we weight

Table 10: Test set classification accuracy on 5 classes (comp.*) of 20 newsgroup dataset for various dictionary sizes (K) and methods.

K	unigram	bigram	trigram	LDA	STC	MedSTC	LCSC	Unified
50	20.15	16.32	19.80	21.69	54.63	76.11	25.68	24.60
100	32.69	14.58	21.59	19.85	60.97	77.70	24.65	22.76
200	47.77	16.62	19.39	26.65	58.72	76.32	34.27	46.65
500	67.16	40.20	25.27	29.51	57.24	75.96	63.32	67.31
1000	69.00	48.49	30.08	36.42	59.80	75.55	68.44	71.71
2000	73.04	53.96	31.30	34.63	55.40	77.14	70.59	74.73
4000	74.53	56.68	34.22	40.67	53.81	76.06	76.27	77.24
8000	74.17	60.92	37.75	29.82	53.20	75.60	78.01	78.06

label predictions. When η is 0, the whole formulation is exactly the same as the previous model, LCSC, except for its estimation procedures (2-blocks coordinate descent of LCSC vs 3-block coordinate descent of the proposed model).

Similar to the classification experiment for LCSC in Section 4.3.4.3, I compare multiclass classification accuracy of various models on popular datasets: WebKB[1] and 20 newsgroup[2] (with standard train-test split). I also compare training time for various β learning procedure against state-of-the-art NMF algorithm, Block Principle Pivoting [25].

5.2.4.1 Classification Peformance

Effect of Number of Topics (K) Similar to a LCSC experiment in Section 4.3.4.3, text classification performance was measured while varying the size of dictionary ($K = [50, \ldots, 8000]$). Except for K, all tunable parameters were fixed as in Section 4.3.4.3 in addition to the newly introduced parameter η being fixing at 10^{-2}.

The unified formulation shows improved performances over the previous model,

[1] Original data is from `http://www.cs.cmu.edu/~webkb/`, but I used a preprocessed dataset from `http://www.cs.umb.edu/~smimarog/textmining/datasets/`

[2] `http://qwone.com/~jason/20Newsgroups/`

Table 11: Comparision of test set classification accuracy for various methods on WebKB4, a subset of 20 newsgroup dataset (comp.*), and the full set

	#class	n-gram	LDA	STC	MedSTC	LCSC	Unified
WebKB	4	88.65	44.25	78.31	86.77	89.37	89.66
comp.*	5	74.53	40.67	60.97	77.70	78.01	78.06
*	20	74.10	34.43	61.14	79.81	80.76	81.23

LCSC, in most situations with the improvement being particularly large in a medium range K: $200 \leq K \leq 2000$. When K is large enough, we obtain smaller benefit. This shows two important new findings. First, the new predictive loss indeed encourages discriminating representations. Second, representations with a limited expression power (small K) benefit larger from the new formulation. This shows the unified model is especially useful when we have a limited data storage.

Compared to all competing methods, the proposed model shows superior performance with enough dictionary size. In contrast, MedSTC model [88], one of the state-of-the-art supervised topic model, showed good performance when K is in medium range, but its maximum performance is still inferior to the proposed model.

Overall Classification Performance Table 11 compares 6 methods including popular models (n-gram and LDA), state-of-the-art unsupervised (STC) and supervised topic model (MedSTC), as well as two models in this dissertation (LCSC and the unified). I tried various configurations and chose the best performance. (n: [1,2,3,4], $K : [50, \ldots, 8000]$, $\lambda, \rho : [10^{-4}, \ldots, 10^{-1}]$, $\eta : [10^{-4}, \ldots, 10^{-2}]$).

On all datasets, the unified formulation outperforms all other competitors, which reveals that employing both sequential and label information induces better classification performance. By comparing two non-supervised methods against supervised methods, STC versus MedSTC or LCSC versus the unified model, I can conclude that incorporating supervised information yields better classification performance.

Table 12: Average training time (sec) of estimating β using Block Principle Pivoting (BPP) algorithm [25] and Greedy Coordinate Descent (GCD) method in Algorithm 2

K	BPP	GCD
50	13.15	9.96
100	22.17	31.24
200	43.93	60.90
500	101.81	276.84
1000	2178.15	706.92
2000	24617.47	1431.19

By comparing two non-sequential methods against sequential methods, STC versus LCSC or MedSTC versus the unified model, I can justify the benefit of utilizing sequential information. In order to maximize classification performance, I recommend employing both labels and sequential information.

5.2.4.2 Training Time Comparison with NMF methods

Section 5.2.3 briefly described the efficiency of the Greedy Coordinate Descent (GCD) method compared to state-of-the-art NMF active-set methods. In their theoretical time complexities, active-set algorithms are highly efficient in standard NMF configurations that have small K situations. In contrast, the GCD algorithm (Algorithm 2) is more efficient in large K situations.

Table 12 presents average β estimation time for Block Principle Pivoting algorithm (BPP) [25], a state-of-the-art active-set-like algorithm, and the proposed GCD algorithm (Algorithm 2). When K is small ($50 \leq K \leq 500$), the BPP algorithm is faster than GCD, but when K is large ($K \geq 1000$), the GCD is faster in large magnitude. Additionally, we can observe the training time for BPP is cubically growing while the one of GCD is linearly growing.

Since both LCSC and the unified model showed their maximum performances in

large K situations (Table 10), the GCD algorithm is much more efficient than active-set algorithms such as BPP. Moreover, combined with good parallelizable properties of the greedy algorithm, the two models works well in practice with GCD.

5.3 Chapter Discussion

This chapter presented a unified way to incorporate both sequential and label information, which achieved even stronger prediction power than LCSC model that only employed sequential information. Improvements were widely observed in various configurations in particular with a limited dimensionality (K) of our representation.

The unified formulation effectively resolves the two challenges that we discussed in the introduction chapter (Chapter 1). On one hand, I effectively solved the sparsity problem by introducing a low-rank matrix factorization and augmenting labels in the formulation. On the other hand, I efficiently modeled text sequentiality by local contexts, which are conditional distributions of words and positioning information. By handling the two challenges properly, the unified model satisfies *good representation* criteria in Section 1.4.

1. Reconstruction Quality: The unified model preserves local word dependencies unlike traditional topic models. The degree of reconstruction quality can be adjusted by coefficients of the model. (See subsequent paragraphs for details).

2. Discriminative Power: We can directly control the discriminative power of resulting representations by increasing or decreasing the predictive loss coefficient η.

3. Interpretability: Similar to LCSC model, we can encourage sparse representations with a large λ in order to obtain more interpretable representations. The unified formulation also provides locally coherent topics as LCSC model does.

4. Computation: We discussed the efficiency of GCD algorithm and found that

GCD is better in our formulation compared to state-of-the-art NMF algorithms. (See Section 5.2.4.2 for details.)

Moreover, similar to LCSC model (Section 4.3), we can directly control the balance of the *good representation* criteria. Each tunable coefficient of (38) handles the balance (copied below).

$$\min_{B,z,D} \sum_{n=1}^{N} \Bigg[\underbrace{\eta f_\Theta(z^{(n)}, y^{(n)})}_{(a)} + \underbrace{\rho \sum_i \|B^{(n)}(i,:) - z^{(n)}\|_2^2}_{(b)} + \underbrace{\|\Phi^{(n)} - DB^{(n)}\|_F^2}_{(c)} + \underbrace{\lambda \sum_i \|B(i,:)\|_1}_{(d)} \Bigg]$$

First, predictive loss (a) directly measures the discriminative power of resulting representations and the corresponding parameter η encourages or discourages the power. Second, (b) and (c) evaluate the reconstruction quality of the model. (b) determines the variability of β in a document, and (c) measures the difference between the model and local observations. A large ρ, which strengthens (b), shrinks β to be close to z, reducing noises and restricting the influence of sequentiality. Third, (d) helps to obtain sparse representations that are useful for interpreting.

So far, I presented various ways to incorporate labels and sequential information in document representations. In the next chapter, I will conclude this dissertation with an overall summary and final remarks.

CHAPTER VI

CONCLUSION

6.1 Thesis Summary

This dissertation addressed two major challenges, sparsity and sequentiality, in document representation learning and made attempts to resolve the issues. I particularly focused on utilizing common supplementary information, labels and sequential information.

First, labels characteristics were examined in Chapter 3. We discussed two new representations that utilize structural proximity and temporal dynamics of labels (emotion annotations). The new representations improved emotion prediction performance and helped us to understand the human emotion further.

Second, in Chapter 4, various levels of sequential dependencies were employed by modeling a joint or condition distribution between words and positions. I examined the joint distribution of spatial and temporal flows of a document in order to model a version-controlled document. The model was useful for measuring sequential and temporal changes of a document as well as predicting abnormal revisions. In addition, the local context model (Section 4.3) captured local distributions of a word with the conditional distribution, which resulted in highly informative and discriminative representations.

Finally, a unified model that utilizes both labels and sequential information was presented in Chapter 5. Following the local context formulation in Chapter 4, I additionally incorporated supervised information during the local representation learning. The new formulation improved the prediction performance further.

As discussed in the introduction, I evaluated presented representations by the

good representation evaluation criteria (Section 1.4). My representations successfully improved all aspects of the criteria. For higher reconstruction quality, additional sequential dependencies were preserved. For stronger prediction capabilities, label characteristics and richer sequential features were utilized. For better interpretability, more compact and sparse representation were examined. For efficient computations, approximation techniques and parallelism were applied.

This dissertation examined a wide range of applications that are applicable in various fields. In Chapter 3, emotion prediction and temporal emotion analysis were covered, which can be applied in psychology or marketing studies to understand social behaviors. In Chapter 4, my systematic methodology that jointly models the content of a document and its history will be helpful for analyzing a large scale collaborate documents such as Wikipedia or GitHub. Local topics and rich local features can be utilized in various text categorization tasks and text summarizations.

6.2 Possible Future Directions

Since I attempted to solve the two most prominent obstacles in document modeling, methodologies in this dissertation would be useful in a wide range of text analysis communities such as machine learning, natural language processing, or information retrieval communities.

An interesting future direction would be generalizing the label characteristics models (Chapter 3) further. In my models, pairwise distances between centroids were preserved in order to preserve structural relationships. Although these models were useful for estimating locations of each labels in the representation space, it does not preserve other statistical relationships such as relative variances. Reflecting relative variances in representations would be helpful when we have large spreading differences between labels. For example, a general emotion term such as `happy` would spread wider than a specific emotion term `peaceful`. Considering such rich relationships

between labels would produce more accurate representations.

Extending kernels (Chapter 4) would be another interesting research direction. I examined spatial or temporal proximities using kernels in this dissertation, but the kernels can easily be extended to examine other proximities such as semantic proximities. For example, coreference relationships can be employed as another type of locality in a kernel. The extended kernel will be based on a monotonically decreasing function based on three proximities: spatial, temporal, and semantical. Additionally, we can learn the kernel directly from our data similar to [53].

6.3 Concluding Remarks

We have entered an era of overwhelming texts. Every books is now being digitized and everybody shares their thoughts in social media. As large scale text analyses have become extremely popular, demands for a *good document representation* have never been greater.

This dissertation proposed efficient methods to learn *good document representations* by utilizing labels and sequential information. By exploiting label characteristics and employing sequential dependencies, I was able to model documents more accurately, make them stronger in prediction, and easier for interpretation. Using approximations and relaxations, learning those representations was largely scalable. I hope this dissertation will draw greater attention to guided document representation learning for the current era of gigantic-scale textual data.

REFERENCES

[1] Beeferman, D., Berger, A., and Lafferty, J. D., "Statistical models for text segmentation," *Machine Learning*, vol. 34, no. 1-3, pp. 177–210, 1999.

[2] Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] Bengio, Y., Schwenk, H., Senécal, J., Morin, F., and Gauvain, J., "Neural probabilistic language models," in *Innovations in Machine Learning*, pp. 137–186, Springer, 2006.

[4] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C., "A neural probabilistic language model," *JMLR*, vol. 3, pp. 1137–1155, 2003.

[5] Blei, D. and Lafferty, J., "Dynamic topic models," in *Proc. of the International Conference on Machine Learning*, 2006.

[6] Blei, D. and McAuliffe, J., "Supervised topic models," *Advances in Neural Information Processing Systems*, 2007.

[7] Blei, D., Ng, A., and Jordan, M., "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[8] Blei, D. M., "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[9] Bonilla, E., Ming, K., Chai, A., and Williams, C., "Multi-task gaussian process prediction," in *NIPS*, 2007.

[10] Čencov, N. N., *Statistical Decision Rules and Optimal Inference*. American Mathematical Society, 1982.

[11] Chen, H., Branavan, S., Barzilay, R., and Karger, D., "Content modeling using latent permutations," *Journal of Artificial Intelligence Research*, vol. 36, no. 1, pp. 129–163, 2009.

[12] Crain, S., Zhou, K., Yang, S., and Zha, H., "Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond," in *Mining Text Data*, pp. 129–161, Springer, 2012.

[13] Das, S. and Chen, M., "Yahoo! for amazon: Sentiment extraction from small talk on the web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[14] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R., "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[15] Dillon, J., Mao, Y., Lebanon, G., and Zhang, J., "Statistical translation, heat kernels, and expected distances," in *Uncertainty in Artificial Intelligence*, pp. 93–100, AUAI Press, 2007.

[16] Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T., "Efficient projections onto the l 1-ball for learning in high dimensions," in *Proc of International Conference on Machine Learning (ICML)*, 2008.

[17] Fodor, I. K., "A survey of dimension reduction techniques," Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002.

[18] Ganu, G., Elhadad, N., and Marian, A., "Beyond the stars: Improving rating predictions using review text content," in *International Workshop on the Web and Databases*, 2009.

[19] Généreux, M. and Evans, R., "Distinguishing affective states in weblog posts," in *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006.

[20] Golder, S. and Macy, M., "Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures," *Science*, vol. 333, no. 6051, pp. 1878–1881, 2011.

[21] Hearst, M. A., "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[22] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Hofmann, T., "Probabilistic latent semantic indexing," in *Proc. of ACM SIGIR Conference*, pp. 50–57, ACM, 1999.

[24] Hong, L., Yin, D., Guo, J., and Davison, B., "Tracking trends: incorporating term volume into temporal topic models," in *Proc. of International Conference on Knowledge Discovery and Data mining*, 2011.

[25] J. Kim, J. and Park, H., "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *International Conference on Data Mining (ICDM)*, IEEE, 2008.

[26] Kennedy, A. and Inkpen, D., "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[27] Keshtkar, F. and Inkpen, D., "Using sentiment orientation features for mood classification in blogs," in *IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 2009.

[28] Koren, Y., "Collaborative filtering with temporal dynamics," in *Proc. of International Conference on Knowledge Discovery and Data mining*, 2009.

[29] Krovetz, R., "Viewing morphology as an inference process," in *Proc. of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 1993.

[30] Lafferty, J. and Lebanon, G., "Diffusion kernels on statistical manifolds," *Journal of Machine Learning Research*, vol. 6, pp. 129–163, 2005.

[31] Lafferty, J., Pereira, F., and McCallum, A., "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proc. of the International Conference on Machine Learning*, 2001.

[32] Larsen, R. J. and Diener, E., "Promises and problems with the circumplex model of emotion," *Review of Personality and Social Psychology*, vol. 13, no. 13, pp. 25–59, 1992.

[33] Le, Q., Jaitly, N., and Hinton, G., "A simple way to initialize recurrent networks of rectified linear units," *arXiv preprint arXiv:1504.00941*, 2015.

[34] Le, Q. and Mikolov, T., "Distributed representations of sentences and documents," in *Proc. of the International Conference on Machine Learning*, 2014.

[35] Lebanon, G., "Axiomatic geometry of conditional models," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1283–1294, 2005.

[36] Lebanon, G., "Information geometry, the embedding principle, and document classification," in *Proc. of the 2nd International Symposium on Information Geometry and its Applications*, pp. 101–108, 2005.

[37] Lebanon, G., *Riemannian Geometry and Statistical Machine Learning*. PhD thesis, Carnegie Mellon University, Technical Report CMU-LTI-05-189, 2005.

[38] Lebanon, G., "Sequential document representations and simplicial curves," in *Proc. of the 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 273–280, AUAI Press, 2006.

[39] Lebanon, G., Mao, Y., and Dillon, J., "The locally weighted bag of words framework for documents," *Journal of Machine Learning Research*, vol. 8, pp. 2405–2441, October 2007.

[40] Lee, D. and Seung, H., "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[41] Lee, H., Raina, R., Teichman, A., and Ng, A., "Exponential family sparse coding with application to self-taught learning.," in *International Joint Conferences on Artificial Intelligence*, 2009.

[42] Lewis, D., Yang, Y., Rose, T., and Li, F., "Smart stop word list bundled with rcv1." `http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop`.

[43] Lewis, D., Yang, Y., Rose, T., and Li, F., "RCV1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[44] Lewis, M. D. and Granic, I., *Emotion, development, and self-organization: Dynamic systems approaches to emotional development*. Cambridge University Press, 2002.

[45] Li, Y. and Osher, S., "Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm," *Inverse Probl. Imaging*, vol. 3, no. 3, pp. 487–503, 2009.

[46] Liu, B., "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, 2012.

[47] Manning, C. D., Raghavan, P., and Schutze, H., *Introduction to Information Retrieval*. Cambridge University Press., 2008.

[48] Mao, Y., Dillon, J., and Lebanon, G., "Sequential document visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1208–1215, 2007.

[49] Mao, Y. and Lebanon, G., "Isotonic conditional random fields and local sentiment flow," in *Advances in Neural Information Processing Systems 19*, pp. 961–968, 2007.

[50] Mao, Y. and Lebanon, G., "Generalized isotonic conditional random fields," *Machine Learning*, vol. 77, no. 2-3, pp. 225–248, 2009.

[51] McCallum, A., Freitag, D., and Pereira, F., "Maximum entropy Markov models for information extraction and segmentation," in *Proc. 17th International Conference on Machine Learning*, pp. 591–598, 2000.

[52] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient estimation of word representations in vector space," *Workshop at International Conference on Learning Representations*, 2013.

[53] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J., "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013.

[54] Mishne, G., "Experiments with mood classification in blog posts," in *Workshop on Stylistic Analysis Of Text For Information Access*, 2005.

[55] Mishne, G. and Maarten, R., "Capturing global mood levels using blog posts," in *AAAI Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[56] Mohammad, S. M. and Turney, P. D., "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 59, no. 000, pp. 1–24, 2011.

[57] Murray, G., Allen, N., and Trinder, J., "Mood and the circadian system: Investigation of a circadian component in positive affect," *Chronobiology international*, vol. 19, no. 6, pp. 1151–1169, 2002.

[58] Na, J., Sui, H., Khoo, C., Chan, S., and Zhou, Y., "Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews," *Advances in Knowledge Organization*, vol. 9, pp. 49–54, 2004.

[59] O'Connor, B., Balasubramanyan, R., Routledge, B., and Smith, N., "From tweets to polls: Linking text sentiment to public opinion time series," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

[60] Pang, B. and Lee, L., "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 2005.

[61] Pang, B. and Lee, L., "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, pp. 1–135, 2008.

[62] Pang, B., Lee, L., and Vaithyanathan, S., "Thumbs up?: sentiment classification using machine learning techniques," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2002.

[63] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M., "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 569–577, ACM, 2008.

[64] Porter, M., "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[65] Quan, C. and Ren, F., "Construction of a blog emotion corpus for chinese emotional expression analysis," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1446–1454, Association for Computational Linguistics, 2009.

[66] Ramage, D., Hall, D., Nallapati, R., and Manning, C., "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proc. of the Conference on Empirical Methods in Natural Language Processing*, 2009.

[67] Roweis, S. and Saul, L., "Nonlinear dimensionality reduction by locally linear embedding.," *Science*, vol. 290, pp. 2323–2326, 2000.

[68] Russell, J. A., "Affective space is bipolar.," *Journal of personality and social psychology*, vol. 37, no. 3, p. 345, 1979.

[69] Russell, J. A., "A circumplex model of affect.," *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[70] Salton, G., *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., 1971.

[71] Shaver, P., Schwartz, J., Kirson, D., and O'connor, C., "Emotion knowledge: Further exploration of a prototype approach.," *Journal of personality and social psychology*, vol. 52, no. 6, p. 1061, 1987.

[72] Strapparava, C. and Mihalcea, R., "Learning to identify emotions in text," in *Proc. of the 2008 ACM symposium on Applied computing*, ACM, 2008.

[73] Strapparava, C. and Valitutti, A., "Wordnet-affect: an affective extension of wordnet," in *Proceedings of LREC*, vol. 4, pp. 1083–1086, Citeseer, 2004.

[74] Teh, Y. W., Jordan, M., Beal, M., and Blei, D., "Hierarchical dirichlet processes," *Journal of the american statistical association*, vol. 101, no. 476, 2006.

[75] Tellegen, A., Watson, D., and Clark, L. A., "On The Dimensional and Hierarchical Structure of Affect," *Psychological Science*, vol. 10, no. 4, pp. 297–303, 1999.

[76] V. Maaten, G. H., "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[77] Verduyn, P., Delvaux, E., Coillie, H. V., Tuerlinckx, F., and Mechelen, I. V., "Predicting the duration of emotional experience: Two experience sampling studies.," *Emotion*, vol. 9, no. 1, p. 83, 2009.

[78] Wallach, H., "Topic modeling: beyond bag-of-words," in *Proc. of the International Conference on Machine Learning*, 2006.

[79] Wand, M. P. and Jones, M. C., *Kernel Smoothing*. Chapman and Hall/CRC, 1995.

[80] Wang, C., Blei, D., and Heckerman, D., "Continuous time dynamic topic models," in *Proc. of Uncertainty in Artificial Intelligence*, 2009.

[81] Watson, D., Clark, L. A., and Tellegen, A., "Development and validation of brief measures of positive and negative affect: the panas scales.," *Journal of personality and social psychology*, vol. 54, no. 6, p. 1063, 1988.

[82] Watson, D. and Tellegen, A., "Toward a consensual structure of mood.," *Psychological bulletin*, vol. 98, no. 2, pp. 219–235, 1985.

[83] Wei, X., Sun, J., and Wang, X., "Dynamic mixture models for multiple time-series.," in *International Joint Conferences on Artificial Intelligence*, 2007.

[84] Weston, J., Bengio, S., and Usunier, N., "Wsabie: Scaling up to large vocabulary image annotation," in *Proc. of International Joint Conferences on Artificial Intelligence*, 2011.

[85] Wiebe, J., Wilson, T., and Cardie, C., "Annotating expressions of opinions and emotions in language," *Language Resources and Evaluation*, vol. 39, no. 2, pp. 165–210, 2005.

[86] Yuan, M. and Lin, Y., "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, 2006.

[87] Zhu, J., Ahmed, A., and Xing, E., "Medlda: maximum margin supervised topic models for regression and classification," in *In Proc. of International Conference on Machine Learning (ICML)*, 2009.

[88] Zhu, J. and Xing, E., "Sparse topical coding," *In Proc. of Uncertainty in Artificial Intelligence (UAI)*, 2011.